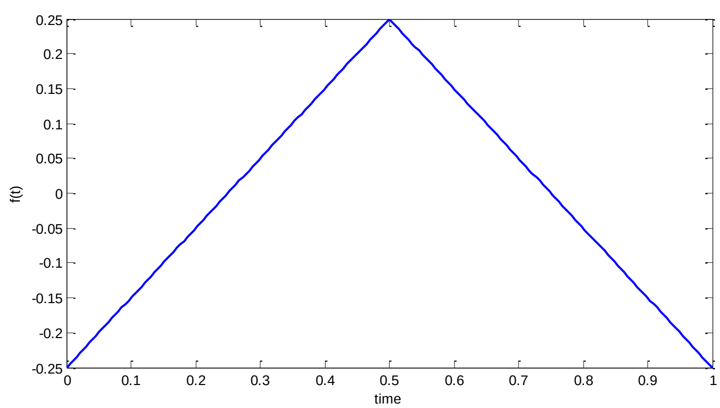
# Fourier Series

1) In class we derived the Fourier coefficients for a square wave. Repeat this analysis for a triangle wave. The exact function is shown below and can be written as $f(t) = t - \frac{1}{4}$ for $0 < t < \frac{1}{2}$ and $f(t) = -t + \frac{3}{4}$ for $\frac{1}{2} < t < 1$. You can follow the procedure in the notes for the square wave to solve this problem. You will need to multiply the given function by sines and cosines and integrate over time from 0 to 1. You can use Wolfram Alpha to help with your integrals. Be sure to write your final result as an infinite sum. i.e. $f(t) = \sum a_n sin\,(2\,\pi n\,t) + b_n \cos(2\,\pi n\,t)$ stating clearly what the coefficients $a_n$ and $b_n$ are. Once you have the coefficients you can plot the Fourier series as a finite sum. Create a plot which superimposes the true function with the Fourier representation where you have only summed the first 3 and then the first 10 modes. You should have excellent agreement by only including 10 modes. For full credit include a short derivation of the coefficients (like we did in class on Wed.), include the final result clearly, and include the requested plot.




2) Write a short MATLAB script to compute the Fourier coefficients for the previous problem numerically. By this we mean rather than carrying out the integration by hand, compute the integrals numerically using the trapezoid rule as we did in class. Represent the function f(t) with 10 data points, 100 data points and 1000 data points. Make a table that compares the first 4 coefficients calculated numerically (for the number of points given as 10, 100, and 1000) versus what they are analytically. The agreement should be very good. Now compute the coefficients using MATLAB's FFT algorithm. State how the FFT result relates to the coefficients you calculated the other two methods. You might need to figure out how MATLAB's FFT algorithm scales the coefficients (maybe by the number of points, maybe a factor of 2 here or there). You can test this by doing a FFT of a pure sine wave. Include the results and the script used to calculate these coefficients numerically in what you hand in. Include a few brief comments on how well the analytical and numerical match.

3) Using the MATLAB data acquisition toolbox (the commands we have been using all semester), your computer's sound card (device name is 'winsound'), and the FFT command – write a program which

takes data from the sound card and computes/displays the power spectrum in "real time". A template for getting started with this task is provided on the website. You should also consult the supplementary notes for the lecture material if you get stuck. The MATLAB template takes ½ second of data and plots the current ½ second of data, the program then runs for ten seconds. Add to this program so that you grab ½ second of data and then plot the current power spectrum. Test the program by whistling and making sure you get a good peak at a single frequency and the peak shifts up and down as change the pitch of your whistle. Include the code and a snapshot of a whistle's transform to show that things are working.

4) Once you get a good peak that seems reasonable, make sure you are getting the peak at the right frequency. You can test the frequency by outputting a known signal from your speaker and measuring through your microphone. You can look at the tutorial for week 1 on the data acquisition toolbox. In that tutorial, we show you how to output data to the speaker through the sound card. You can setup, within the same program, an analoginput and analogoutput. You can start them sequentially in your program and thus simultaneously output through your speaker and measure through the microphone. To test this idea, you should first output a simple sine wave at a few hundred Hz and make sure that your power spectrum is computing a peak at the known frequency. If this works, output a triangle wave (use the command `sawtooth(2*pi*t*w,0.5)` instead of `sine(2*pi*t)` to create the triangle wave at a frequency w). Look at the measured power spectrum and look at the frequencies where you see peaks in the power spectrum. The peaks should be located at frequencies which correspond to those computed in part one of this week's assignment. The relative magnitude of the peaks in experiment will likely not quite match the theory because the speaker is very cheap and distorts the signal significantly. When you turn this assignment in, hand in your code (it should be documented/commented) and your results. Your result should show the measured spectrum of a triangle wave and have some commentary on whether the measured peaks are in the right location and how well the experiment matches what we would expect.