**CSC 745 Advanced Multimedia Programming**

**Exercise: Calculate zero crossings for a signal**

**Goals**

Learn to calculate zero crossings for an audio signal

**Background**

Audio signals can be characterized in a number of different ways. Often sophisticated processing is done – short term FFT, mel-cepstral frequency coefficients, linear predictive coding (LPC) coefficients, etc. But it’s also possible to use (or include) some simple features that are easy to calculate. One example is zero-crossings. Zero-crossings count the number of times the signal crosses the zero line, and gives a rough approximation of frequency – consider, for example, how many times in each cycle a sine wave crosses the zero line. The signal is often differentiated before calculating these features so we’ll include the option of doing that.

**Procedure**

1. Download the exercise description from Blackboard.
2. Activate your audio virtual environment and open a new Python file, ExZeroCrossings.py in your IDE or text editor.
3. Import numpy and modules for a) reading and writing a wav file; b) playing audio stored in a numpy array. I’ll import scipy.io.wavfile and sounddevice (as sd).
4. Read an audio file, such as *birch\_canoe.wav,* into a numpy array.
5. Complete the difference() function to differentiate the signal. You might want to focus on a segment – samples ranging from 16000 to 40000
6. You might want to play the original and differentiated signals.
7. Optionally, scale the audio to use the full dynamic range.
8. Complete the function *calc\_zc(),* returning the zero-crossings envelope. I suggest you use a 20 ms analysis window for both, although you should experiment with other values (for 20 ms, divide the sample rate by 50 – that gives the number of samples in a 20 ms window).
9. Complete the plot\_wave2() function to plot two signals. Plot the sound and the zero-crossings.
10. Does the zero-crossing signal look different for the original sound vs the differenced sound? If so, why do you think it looks different?

**Deliverables**

Submit your .py file on Blackboard before the due date.