Before the concept of cepstral peak prominence measure, there existed many conflicting ideas in the research of voice quality. Hillenbrand came up with several measures in his 1994 paper describing possible ways of quantifying voice quality. Cepstral peak prominence was shown as one of the many. Through its calculation, it makes the original sound wave structured to be highly periodic and therefore allows the easier detection of the peak. The computation of CPP is as follows: 1. Take the Fourier transform of the original sound wave, we thus obtain the spectrum of the sound wave. 2. We now take the log of base 10 of the spectrum with respect to its y-axis and we take its absolute value. 3. We now take the inverse Fourier Transform of the result of step 2, and we call the result cepstrum. Note that we should get back seconds and amplitude for units(except they do not represent anything in the physical sense and thus we name them quefrency and that The quefrency is a measure of time, though not in the sense of a signal in the time domain). 4. Using linear regression, we try to fit a line through our result of 4. 5. Finally, CPP is computed by taking the difference of peak of the cepstrum and corresponding y-value of the fit line. I was first looking over some examples of correctly calculated CPP online for some audio found them to be very useful for analyzing voice quality.

The goal of this project is to investigate and study the CPP measure and its relationship with voice quality. The coding process for this project was relatively easy since the MATLAB version code was already provided to me. Then the MATLAB code could be used as pseudo-code for my implementation. However, there was one thing that I needed to change for testing, I had to change the “variables” variable to its corresponding values to my parameters, and make the function an independent function not requiring voicesauce.

Now for testing results, it’s an update from last time where voicesauce failed to work in my device. I was able to obtain the vectors of fundamental frequency for my sample audio. However, the function was failing to work. I think the cause of the problem is that the MATLAB version of the CPP had something inherent to do with voicesauce. I was not able to do a careful line-by-line examination of the code as to which line was the reason in time.

The key lessons learned for this project are mainly: 1. The transcription from MATLAB to python was fun since you see the linkage between MATLAB and numpy in there. 2. If computed correctly, CPP is a great measure for voice quality analysis.

As for the remaining challenges, I do want to continue working on the CPP function until it compiles successfully on some audio files that others have correctly computed on. After that, if it’s possible I could work on the incorporation of my CPP function for voicesauce.

Reference:

Hillenbrand, James, et al. “Acoustic Correlates of Breathy Vocal Quality.” *Journal of Speech, Language, and Hearing Research*, vol. 37, no. 4, 1994, pp. 769–778., doi:10.1044/jshr.3704.769.

“PowerCepstrum.” *PowerCepstrum: Get Peak Prominence...*, www.fon.hum.uva.nl/praat/manual/PowerCepstrum__Get_peak_prominence___.html