Rose Coloured Waveform (RCW)

Resources / Links

Carroll College Library & Learning Commons - The Fourier Transform Applied to Sound
https://www.youtube.com/watch?v=mQzJ_1Rj6qo

- "Any continuous periodic function can be expressed in terms of sines and cosines"
- Fourier transform is *invertible.*
- FT takes position over time data/ position over time functions, and turns them into magnitude over frequency functions.
- "… take the Fourier transform of your audio file, diminish the unwanted frequencies, invert the transform and you're left with the original audio file just without those unwanted frequencies."
- "How do we get from position over time to frequency over time?"
  - If we take the Fourier transform of an audio file, we wouldn't see the different notes at different times, we would see all the notes at once.
  - The solution is to split the audio file into a bunch of little segments at different times and take the Fourier transform of each of those segments. Then string the transforms together in the same order that the segments were taken, and we're left with what frequencies are present and each moment.

Jeff Heaton - Extract Musical notes from Audio in Python with FFT
https://www.youtube.com/watch?v=rj9NOiFLxWA
https://github.com/jeffheaton/present/blob/master/youtube/video/fft-frequency.ipynb

- The size of the array going into FFT is the same size as the array coming out of FFT. This is the number of "buckets" for frequencies.
- How many times was each frequency used throughout the complex waveform?
- The video creator does two passes over the waveform. The first finds the maximum amplitude so know how to scale the frequencies, and the second produces his animation.

3Blue1Brown - But what is the Fourier Transform? A visual introduction.
https://www.youtube.com/watch?v=spUNpyF58BY

- Theoretical explanation of FT. Useful without points to note.

The Coding Train - Sound Visualization: Frequency Analysis with FFT
https://www.youtube.com/watch?v=2O3nm0Nvbi4

- Useful visualization of "buckets" of frequencies.

Valerio Velardo - How to Extract the Fourier Transform with Python
https://www.youtube.com/watch?v=R-5uxKTRjzM

- How to utilize FT to transform waveform samples into frequency magnitude

Valerio - Velardo - Short-Time Fourier Transform Explained Easily
https://www.youtube.com/watch?v=-Yxj3yfvY-4

- Declare a "frame size" and take various snapshots of the audio file and then utilize DFT on each frame to analyze an extended waveform.
- (32:16) formula for visualizing sound into a spectrogram

Valerio - Velardo - How to Extract Spectrograms from Audio with Python
https://www.youtube.com/watch?v=3gzI4Z2OFgY
https://github.com/musikalkemist/AudioSignalProcessingForML/blob/master/16%20-%20Extracting%20Spectrograms%20from%20Audio%20with%20Python/Extracting%20Spectrograms%20from%20Audio%20with%20Python.ipynb

- How to use STFT to create a spectrogram from a waveform

Steve Brunton - Denoising Data with FFT [Python]
https://www.youtube.com/watch?v=s2K1JfNR7Sc

- Demonstration of adding random noise to a set with two unique sine waves, denoising by removing all frequencies that do not reach a designated power level, and inversely transforming the data to reproduce the original set before the random noise was added.

Scale the real part of complex numpy array
https://stackoverflow.com/questions/13567089/scale-the-real-part-of-complex-numpy-array

- Nparray.real *= nparray

Short-Time Fourier Transform(Advanced Signal Processing Toolkit)
https://www.ni.com/docs/en-US/bundle/labview-advanced-signal-processing-toolkit-api-ref/page/lvasptconcepts/aspt_stft.html#:~:text=Therefore%2C%20the%20STFT%20is%20a,results%20of%20windowed%20Fourier%20transforms.

- "Therefore, the STFT is a function of time and frequency that indicates how the spectral content of a signal evolves over time. A complex-valued, 2-D array called the STFT coefficients stores the results of windowed Fourier transforms."
- "The window length also affects the time resolution and the frequency resolution of the STFT. A **narrow window** results in a fine time resolution **but** a coarse frequency resolution because narrow windows have a short time duration but a wide bandwidth. A **wide window** results in a fine frequency resolution **but** a coarse time resolution because wide windows have a long time duration but a narrow frequency bandwidth. This phenomenon is called the window effect. *You cannot obtain a fine time resolution and a fine frequency resolution simultaneously using the STFT*. With a time-invariant window, the STFT has the same time resolution and frequency resolution across the entire time-frequency plane."

Library documentation

librosa.fft_frequencies()
https://librosa.org/doc/0.10.1/generated/librosa.fft_frequencies.html#librosa.fft_frequencies

librosa.stft()
https://librosa.org/doc/0.10.1/generated/librosa.stft.html#librosa.stft

librosa.istft()
https://librosa.org/doc/0.10.1/generated/librosa.istft.html#librosa.istft

scipy.signal.check_COLA()
https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.check_COLA.html

soundfile.Soundfile.read()
https://pysoundfile.readthedocs.io/en/latest/#soundfile.SoundFile