FREQUENCY SPACE AND FOURIER TRANSFORMS

Eric Prebys

Discrete Transformations

- In general, we can express any periodic function as sum of sines and cosines

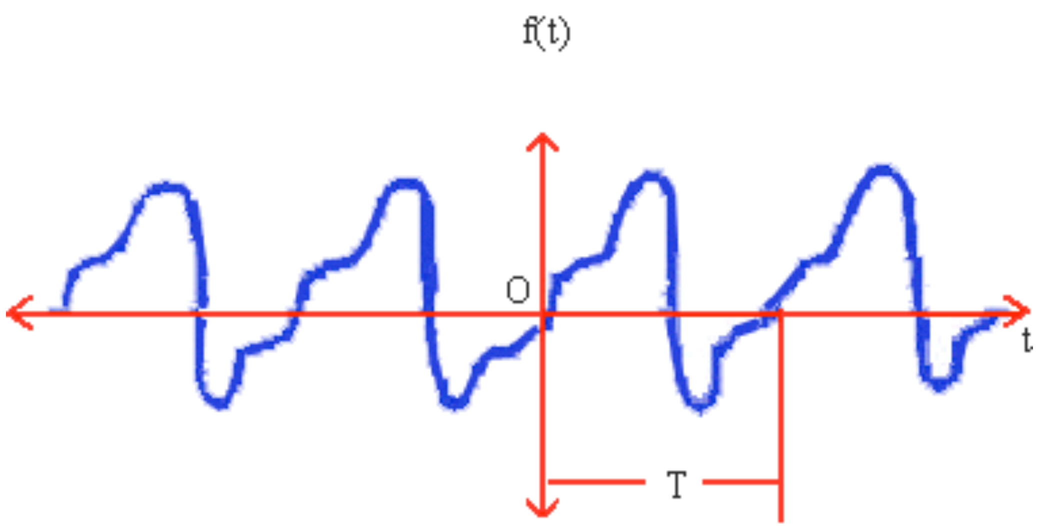


Fig. 7

$$f(t+T)=f(t)$$

$$=\sum_{n=0}^{\infty} C_n \cos\left(\omega_n t+\delta\right)$$

$$f=a_0+\sum_{n=1}^{\infty}\left(a_n \cos\omega_n t+b_n \sin\omega_n t\right)$$

$$\omega_n=\frac{2\pi n}{T}$$

Fourier Series

Reminder: Sum and Product Rules

$$
\boxed{
\begin{aligned}
\sin(\alpha+\beta) &= \sin\alpha\ \cos\beta + \cos\alpha\ \sin\beta \\
\cos(\alpha+\beta) &= \cos\alpha\ \cos\beta - \sin\alpha\ \sin\beta \\
\sin(\alpha-\beta) &= \sin\alpha\ \cos\beta - \cos\alpha\ \sin\beta \\
\cos(\alpha-\beta) &= \cos\alpha\ \cos\beta + \sin\alpha\ \sin\beta
\end{aligned}
}
$$

$$
\int_0^T \cos(j\omega t)\cos(k\omega t)\,dt = \frac{1}{2}\int_0^T \left[\cos\left((j+k)\omega t\right)+\cos\left((j-k)\omega t\right)\right]dt
$$

$$
= \frac{T}{2}\delta_{j,k}
$$

Extracting the Coefficients

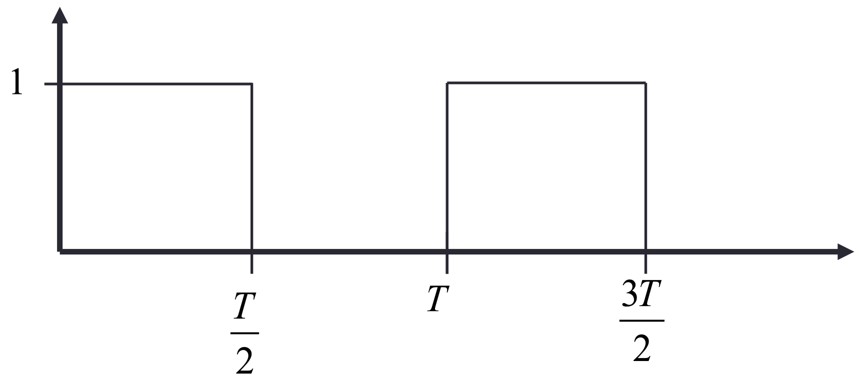
$$\int_0^T f(t)\,dt = a_0 T \rightarrow a_0 = \frac{1}{T}\int_0^T f(t)\,dt$$

$$\int_0^T f(t)\cos\omega_n t\,dt = \int_0^T a_n \cos^2 \omega_n t\,dt = T\frac{a_n}{2} \rightarrow a_n = \frac{2}{T}\int_0^T f(t)\cos\left(\frac{2\pi n}{T}t\right)dt$$

$$\int_0^T f(t)\sin\omega_n t\,dt = \int_0^T b_n \sin^2 \omega_n t\,dt = T\frac{b_n}{2} \rightarrow b_n = \frac{2}{T}\int_0^T f(t)\sin\left(\frac{2\pi n}{T}t\right)dt$$

Fourier Transform

Example: Square Wave



$$0 \le t < \frac{T}{2}: \quad f(t) = 1$$

$$\frac{T}{2} \le t < T: \quad f(t) = 0$$

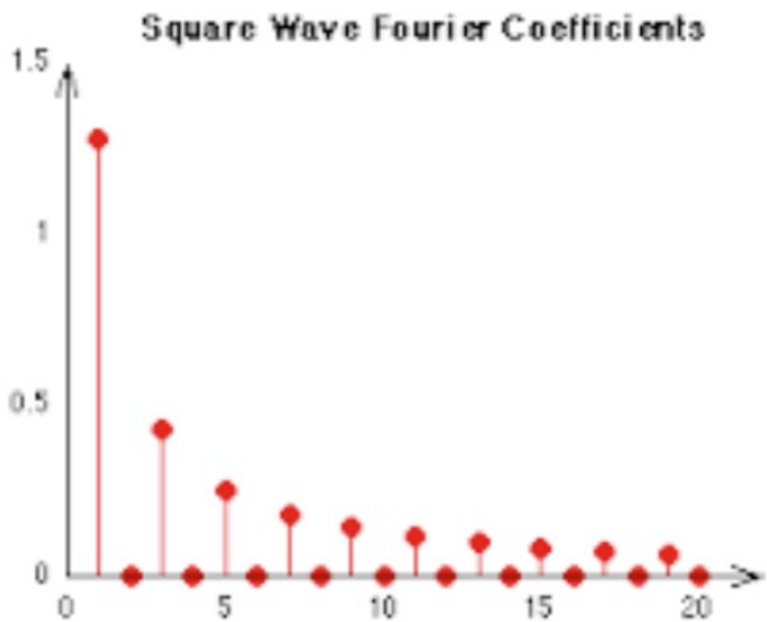
$$a_0 = \frac{1}{T}\int_0^T f(t)\,dt = \frac{1}{T}\int_0^{T/2} dt = \frac{1}{2}$$

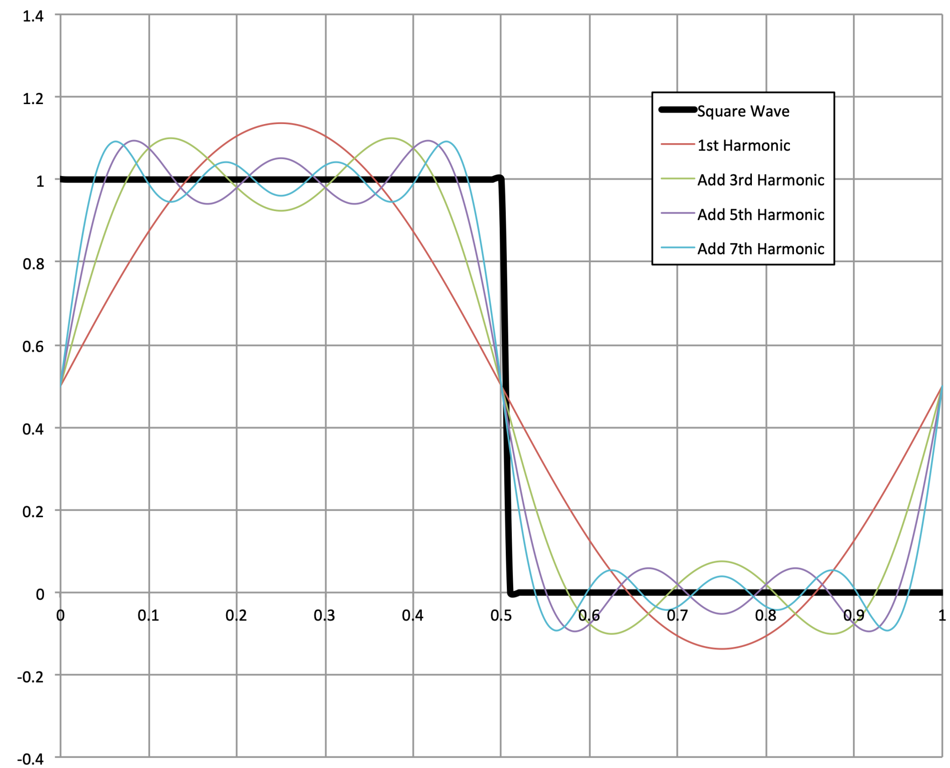
$$a_n = \frac{2}{T}\int_0^T f(t)\cos\left(\frac{2\pi n}{T}t\right)dt = \frac{2}{T}\int_0^{T/2}\cos\left(\frac{2\pi n}{T}t\right)dt = \frac{1}{\pi n}\sin\left(\frac{2\pi n}{T}t\right)\Bigg|_0^{\frac{T}{2}} = 0$$

$$b_n = \frac{2}{T}\int_0^T f(t)\sin\left(\frac{2\pi n}{T}t\right)dt = \frac{2}{T}\int_0^{T/2}\sin\left(\frac{2\pi n}{T}t\right)dt = -\frac{1}{\pi n}\cos\left(\frac{2\pi n}{T}t\right)\Bigg|_0^{\frac{T}{2}} = \begin{array}{l} 0 \ (n \text{ even}) \\ \dfrac{2}{\pi n} \ (n \text{ odd}) \end{array}$$

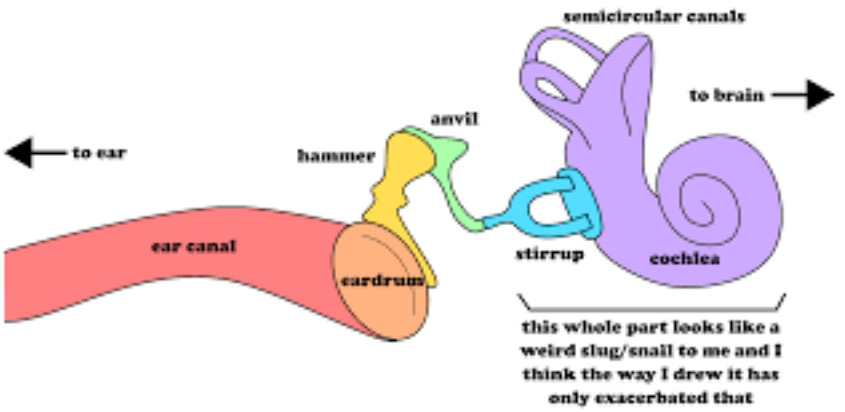
Example: Square Wave

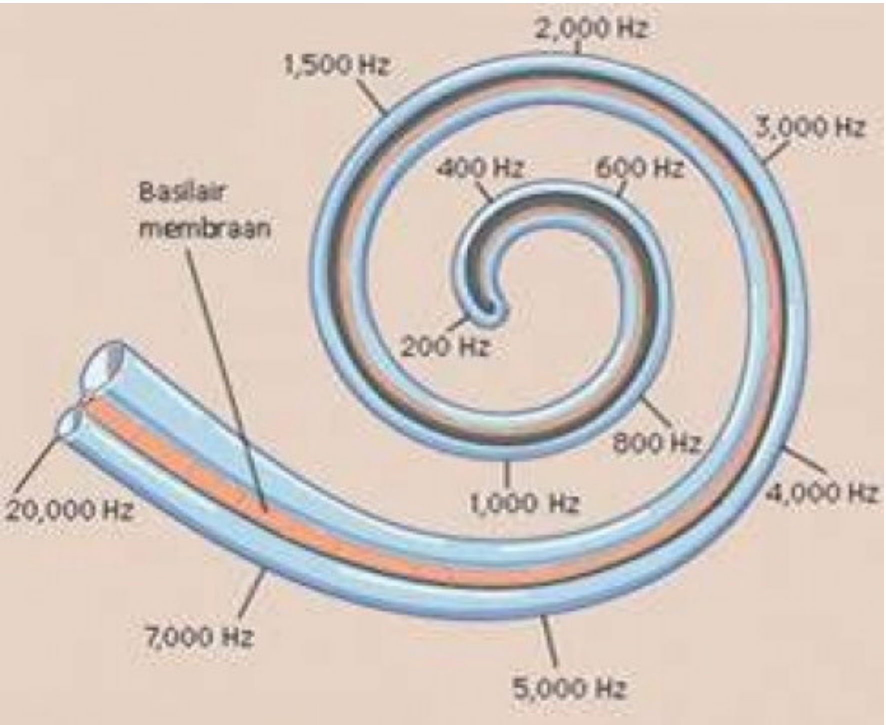
$$f(t) = \frac{1}{2} + \sum_{n=1,3,5\ldots}^{\infty} \frac{2}{n\pi} \sin\left(\frac{2\pi n}{T} t\right)$$

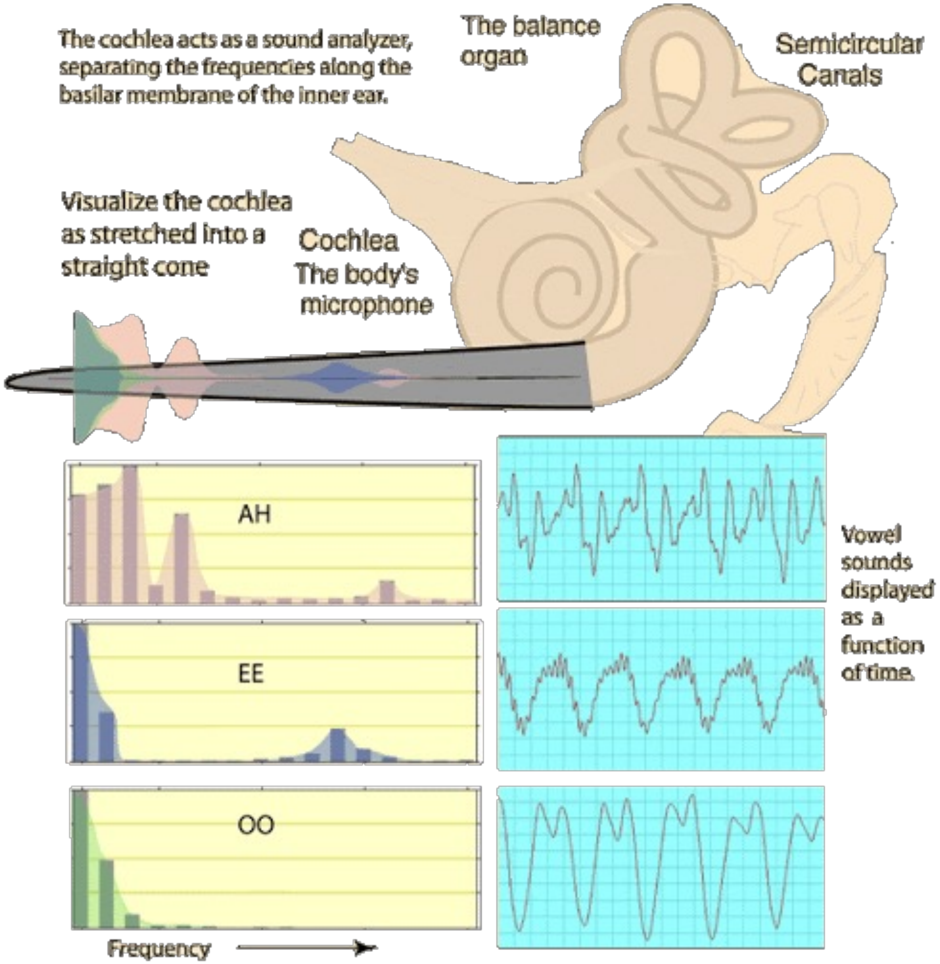




The Ear and Fourier Transformations

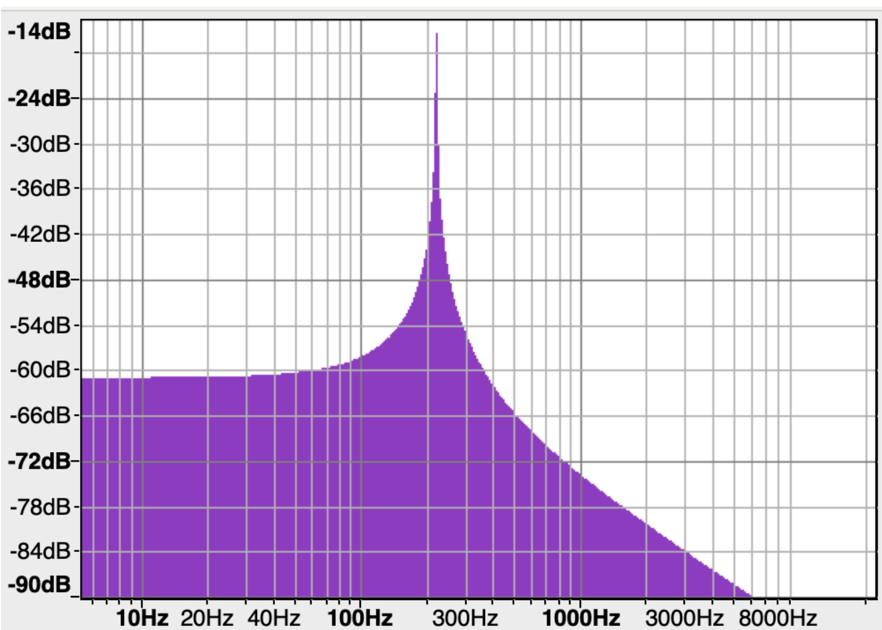




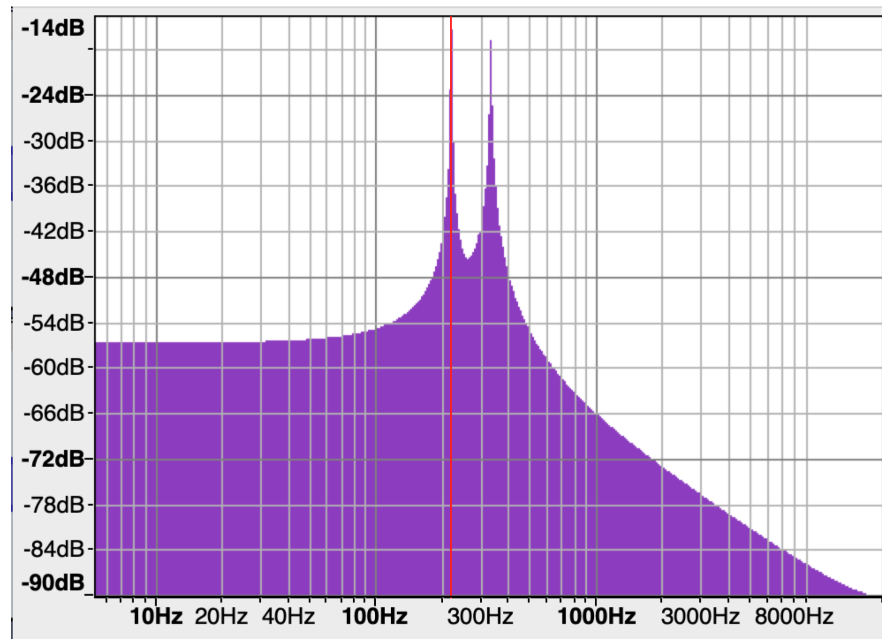


Perception of Hearing

- Because of the way the ear works, we do not directly sense the motion of the ear drum, but we do sense the individual frequency components of the sound, with very high precision.
  - This allows us to hear individual notes:

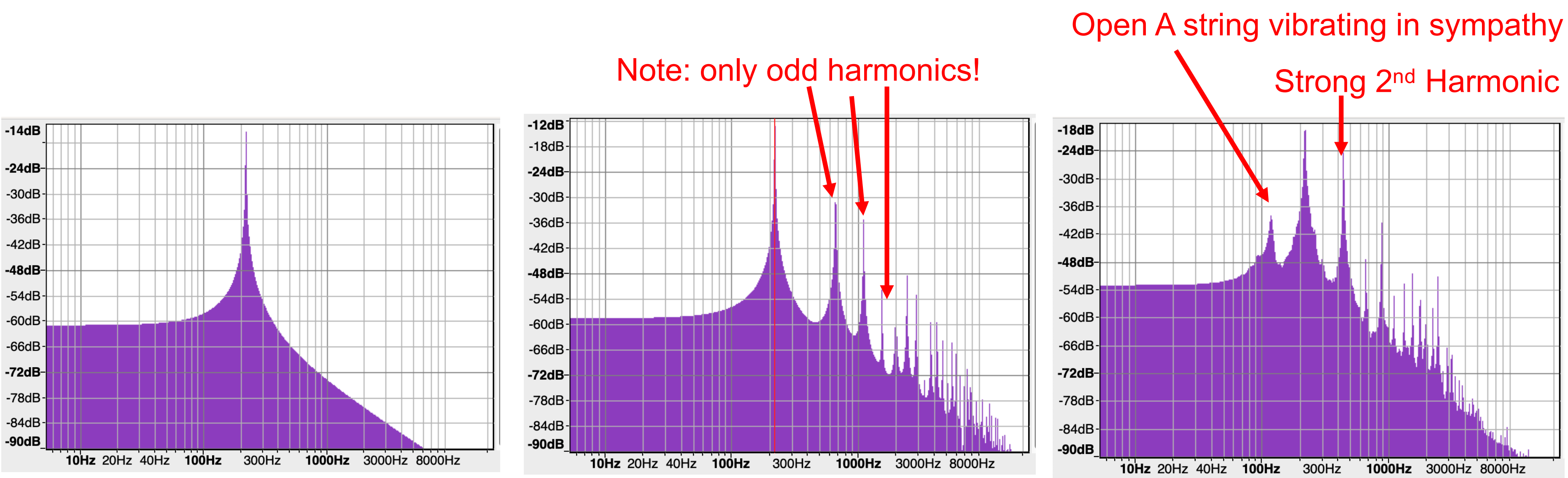


A_3 (220 Hz)



Perfect 5th (220Hz+330 Hz)

- Spectral content (overtone series) also determines how a note sounds to us.



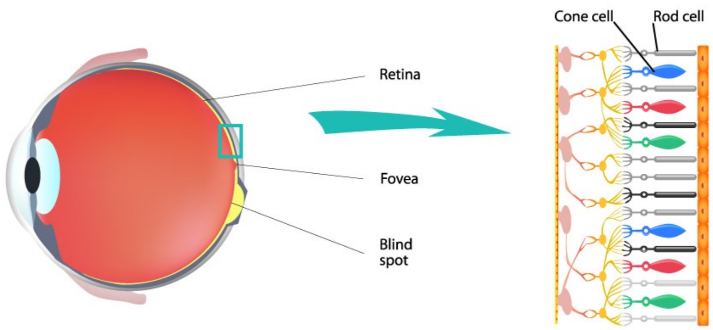
A_3 (220 Hz sine wave)

220 Hz square wave

A_3 on Les Paul played through a really lousy amp.

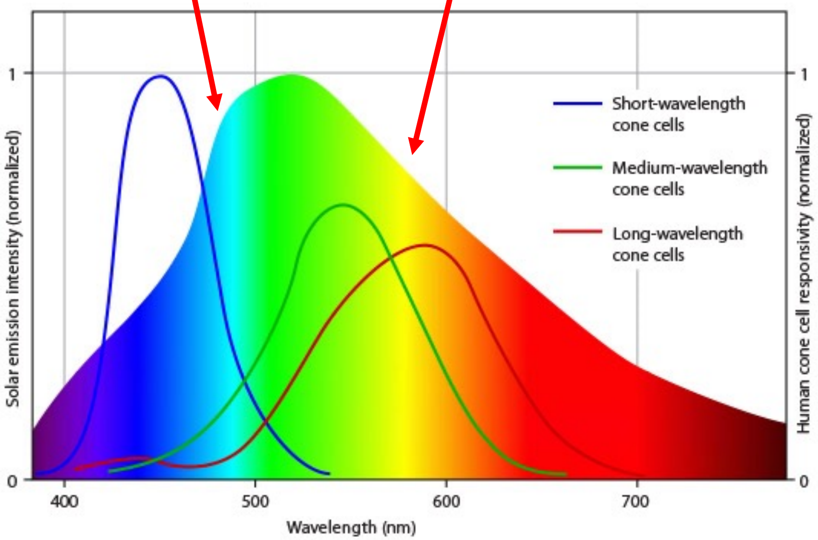
Digression: How We See Color

- Our perception of color is also based on frequency, but unlike sound, we don't distinguish individual colors.
- For each point, we perceive a *single* color based on the relative stimulation of the three types of color-sensitive "cones" in our retinas.



blue+green=cyan

red+green=yellow



"red" cones are also sensitive to violet, thus red+blue=violet

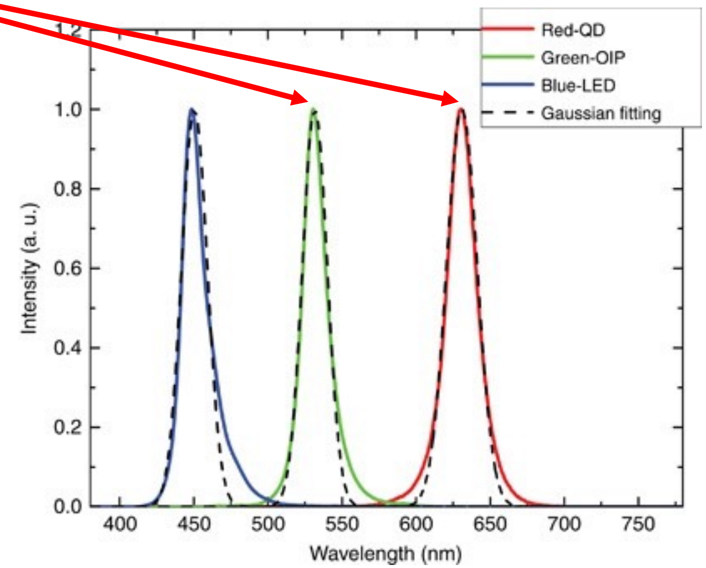
red+green+blue=white

- Projectors and displays mimic colors using combinations of red, green, and blue.

No image shows ”true” colors

- Example: If I take the from a helium discharge tube and put it through a diffraction grating, I’ll see well defined lines corresponding to individual wavelengths

- On the other hand, if I take the color “yellow” that I’m projecting here and pass it through a diffraction grating, I’ll see the spectra of the green and red LEDs



- Fish and some birds have a fourth shorter wavelength cone, so they would not see an RGB projection of violet as the same as pure violet.

Complex Representation

$$\cos\omega_n t = \frac{\left(e^{i\omega_n t} + e^{-i\omega_n t}\right)}{2}$$

$$\sin\omega_n t = \frac{\left(e^{i\omega_n t} - e^{-i\omega_n t}\right)}{2i}$$

$$a_n\cos\omega_n t + b_n\sin\omega_n t = \left(\frac{a_n}{2} + \frac{b_n}{2i}\right)e^{i\omega_n t} + \left(\frac{a_n}{2} - \frac{b_n}{2i}\right)e^{-i\omega_n t}$$

$$= c_n e^{i\omega_n t} + c_{-n} e^{i\omega_{-n} t}$$

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\omega_n t}$$

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t)e^{-i\omega_n t}\,dt$$

If result is real, then

$$c_{-n} = c_n^*$$

Power in the Fourier spectrum

- We're generally interested in average power as a function of frequency, and in general

$$P \propto \left\langle v^2 \right\rangle$$

- We need to be careful when dealing with nonlinear functions of complex representations, because in general

$$\mathrm{Re}\left\{A\right\}^2 \neq \mathrm{Re}\left\{A^2\right\}$$

Power (cont'd)

- In our frequency representation

$$v(t)=\sum_{n=-\infty}^{\infty} c_n e^{i\omega_n t}$$

$$\langle v^2(t)\rangle = \left\langle \left(\sum_{n=-\infty}^{\infty} c_n e^{i\omega_n t}\right)^2 \right\rangle$$

$$= c_0^2 + 2\sum_{n=1}^{\infty}\langle c_n c_{-n}\rangle + \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty,\neq i}\langle c_i c_j e^{i(\omega_i+\omega_j)t}\rangle$$

0

$$c_{-n} = c_n^*$$

$$\langle f(t)\rangle = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\,dt$$

$$= c_0^2 + 2\sum_{n=1}^{\infty}|c_n|^2$$

$$= \sum_{n=0}^{\infty}|s_n|^2$$

- In other words, the power spectrum has the same spectral content as the amplitude spectrum

Discrete Fourier Transformation

- If we have *N* time domain samples over a time *T*, the time between samples will be

$$\Delta t = \frac{T}{N}$$

- This will correspond to *N* frequency domain, with a frequency step

$$\Delta f = \frac{1}{T}$$

- That is, the minimum frequency would be one in which the period is the entire time of acquisition.
- The maximum frequency would be

$$N\Delta f = \frac{N}{T} = (\text{sample rate})$$

- Nyquist's theorem tells us we can only use up to half of this frequency, to the extra information is for phase values.

Fast Fourier Transforms (FFTs)

- In order to find the Fourier Coefficients using integration, we would have to integrate *N* time domain points for each of the *N* frequency values, so the computation time would go as N^2
- Luckily, there's an algorithm called "Fast Fourier Transform", for which the time goes like log(*N*).
- The FFT tool in Python is np.fft.fft.
  - In general, this takes *N* complex time domain values and returns *N* complex frequency domain values, corresponding to

$$c_{(-\frac{N}{2})} \rightarrow c_{(+\frac{N}{2})}$$

  - Because we usually deal with real values in the time domain

$$c_{-n} = c_n^*$$

  - Since we're generally only interested in the positive values, the the returned values are in the order

$$c_0, c_1, c_2, \ldots, c_{(\frac{N}{2})}, c_{(-\frac{N}{2})}, \ldots, c_{-2}, c_{-1}$$