

1 | base knowlege

1.1 | primitive root of unity

1.1.1 | **a number r is a primitive n th root of unity iff n is the smallest counting number for which $r^n = 1$.**

1.1.2 | <https://mathworld.wolfram.com/PrimitiveRootofUnity.html> **source**

1.2 | convolution theorem

1.2.1 | **'depends fundamentally on the convolution theorem, which provides an efficient way to compute the cyclic convolution of two sequences. It states that the cyclic convolution of two vectors can be found by taking the discrete fourier transform of each of them, multiplying the resulting vectors element by element, and then taking the inverse discrete fourier transform.'**

2 | sources

2.1 | **explanation of multiplication algorithm**

2.2 | **paper explaining the multiplication ANDO EMERENCIA (S1283936)**

2.3 | **FFT Medium Blog Post**

2.4 | **wikipedia on schonhage-strassen (multiplication algo)**

3 | uses of FFT

3.1 | **convert mixed signals into constituent sinusoids**

3.2 | **multiply polynomials using convolution theorem**

3.3 | **reduce matrix dimensionality**

3.4 | **audio processing (eg. bass boost, or radio denoising for eg. wifi)**

3.5 | **MRI machines? scan certain parts using different overlapping sinusoidal magnitudes of magnetic field**

3.6 | **microscope or astronomy image decomposition**

3.7 | **connection to Heisenberg uncertainty principle**

3.7.1 | **the fourier transform graph says something about correlation**

1. eg. for a pure signal (sine 5Hz), Ξ

4 | **3b1b video** <https://www.youtube.com/watch?v=spUNpyF58BY>

4.1 | **unmixing waves**

4.1.1 | **the added up ones seem needlessly complex for such a little amount of info**

4.2 | **rotating the wave around a circle**

4.2.1 | **aka: wave around the circle is polar coords: length = magnitude of wave at that point, offset = phase + some angular velocity (the 'rotation' frequency)**

4.2.2 | **there are two frequencies: 1. the frequency at which the vector goes around the circle 'winding frequency', and 2. the original and 'true' frequency of the wave**

4.2.3 | **when the frequencies match, all the high points are on the right and low points are on the left... question is how can we quantify this specialness**

4.2.4 | **center of mass as a function of the winding frequency**

1. frequency of zero is high, and then it wobbles for a while until a frequency matches

4.3 | **central construct**

4.3.1 | **original plot (intensity | time)**

4.3.2 | **winding chart (wound signal | signal, winding frequency)**

4.3.3 | **center-of-mass plot (x coord | winding frequency)**

1. the spike at zero only happens because the original freq doesn't oscillate about zero

4.4 | **he calls this the 'almost Fourier transform'**

4.4.1 | **additive: you can take the almost fourier transform first or you can take the sum first and you will get the same center-of-mass plot out**

1. pause and ponder: multiple arrows going around the circle, tip to tail

4.5 | **formalizing the 'center of mass'**

4.5.1 | **complex numbers: works well for 2d plane and rotation can be described by**

$$e^{2\pi i t}$$

by multiplying that t by a scalar, you can change the frequency:

$$e^{2\pi i f t}$$

4.5.2 | **actual formalization**

1. convention: rotate in clockwise direction

$$e^{-2\pi i f t}$$

2. let the original function be called $g(t)$, then scale by that for the 'vector following the original graph magnitude'

$$g(t)e^{-2\pi i f t}$$

3. tracking 'center of mass': sample points and average them
if N is the number of points that you sample and t_k is the k-th sampled point,

$$\frac{1}{N} \sum_{i=1}^N g(t_k) e^{-2\pi i f t_k}$$

4. and if we want a more accurate sample, just take the limit to infinity

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N g(t_k) e^{-2\pi i f t_k}$$

5. which is really the same as taking the integral

$$\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} g(t) e^{-2\pi i f t} dt$$

6. but we don't actually need to divide by the time interval

$$\int_{t_1}^{t_2} g(t) e^{-2\pi i f t} dt$$

This means that when a frequency persists for a long time, it gets scaled more