Fourier Transforms – Introduction

CS370 Lecture 17 – Feb 10, 2017

Today – Intro to Fourier Transforms

- Point out some applications.
- Describe the basic premise of Fourier transforms
- Examine the math for the *continuous* Fourier series of a function.

Introduction to Fourier Analysis

The ***Fourier transform*** applied to continuous functions or discrete data can provide useful insights and tools for many applications.

Examples include:

- “signal processing” in general (filtering, denoising, etc.)
- sound/audio manipulation.
- image/video processing (e.g. JPEG-style compression).
- analyzing sensor data.
- electromagnetic signals.
- lots more!

Appendix G discusses CT Scanning.

Fourier Analysis – Basic Idea

1) Transform some function/data into a different representation that emphasizes/reveals **frequency** of information in the data.

2) Perform processing or analysis in this “frequency domain” representation, which makes certain tasks easier.

3) Transform back again.

The original data/function is said to be in the…

- “time domain” if f is a function of time, $f(t)$.
- “spatial domain” if f is a function of space/position, $f(x)$.

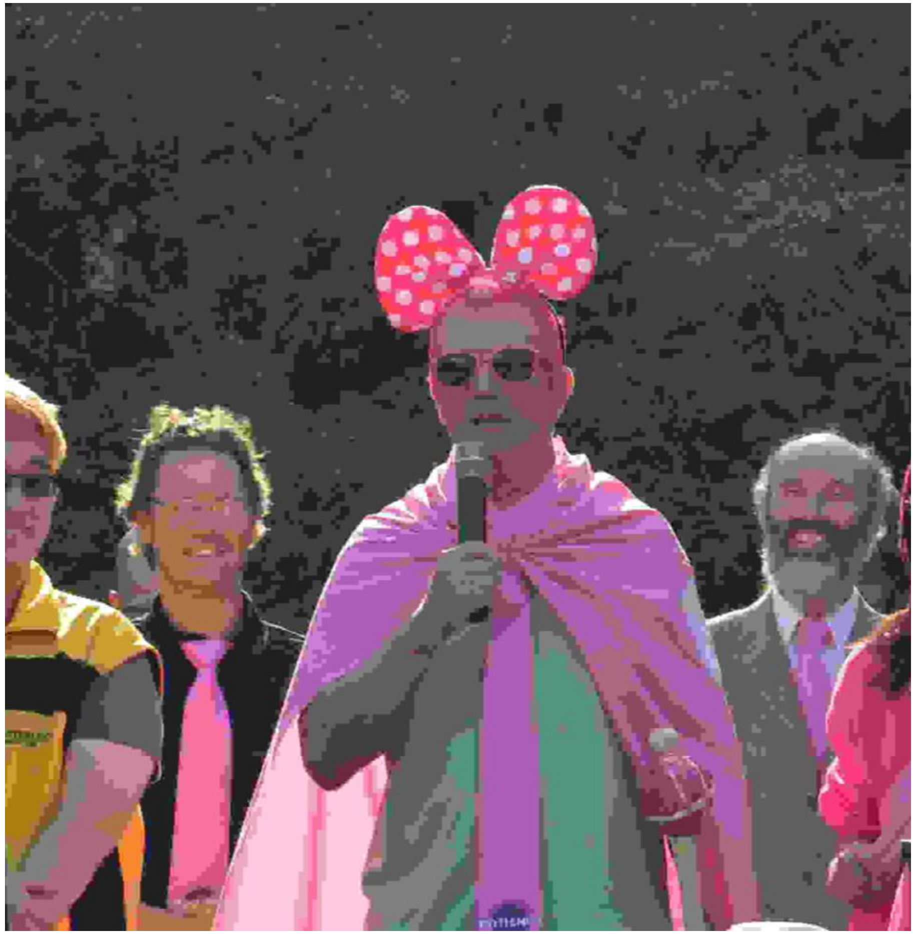
Example: Image Compression

Original Bitmap: 46MB
High Quality JPEG: 4.5 MB

Low Quality JPEG: 170 KB

Example: Image Compression – Zoom!

Original - Zoomed



Low Quality JPEG - Zoomed

Example: Image Compression – Super Zoom!

Original - Zoomed

Low Quality JPEG - Zoomed

Example: Image Compression

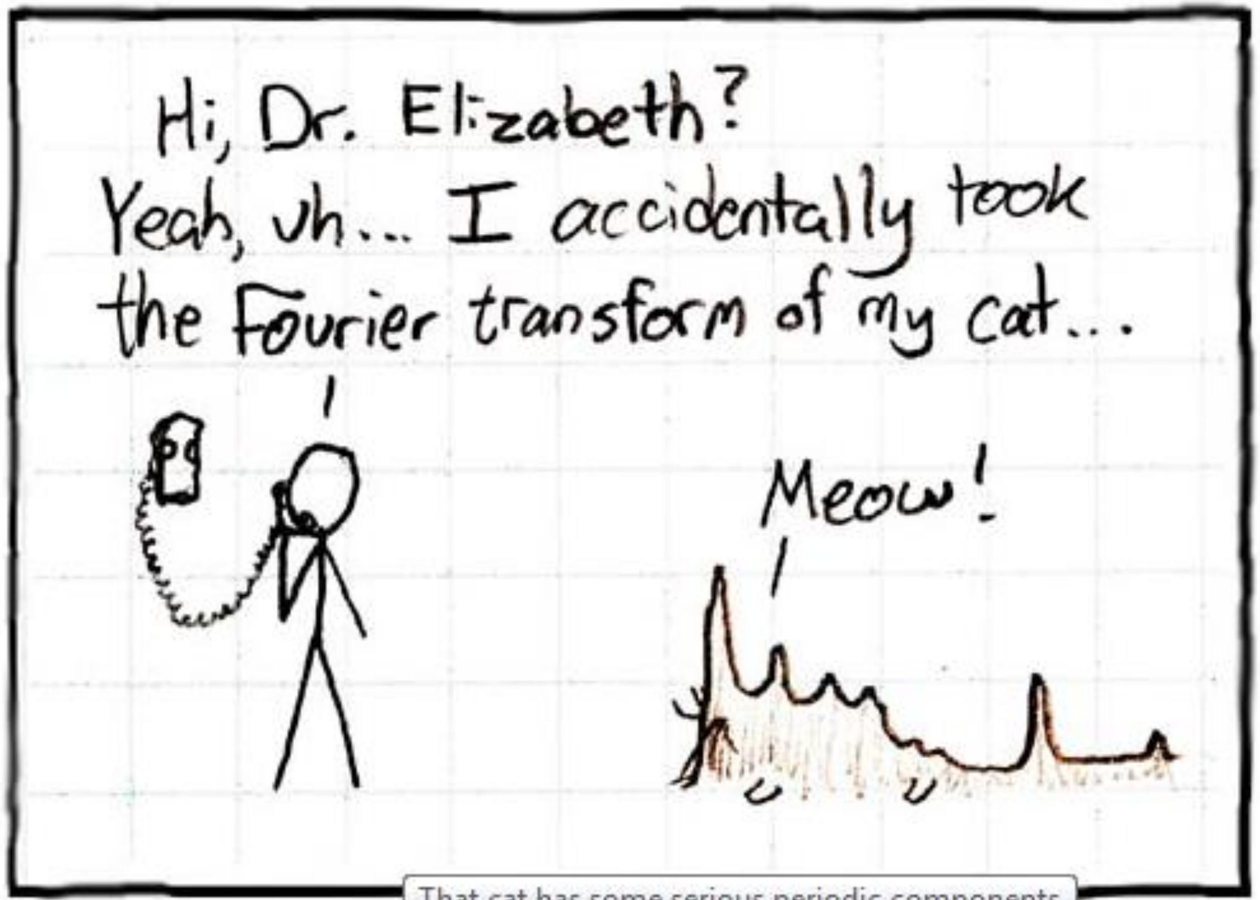
JPEG and other *lossy* image compression formats typically use Fourier analysis and related principles.

Common image compression errors relate to aspects of Fourier analysis. (And we'll see why.)

e.g. "Ringing" effects.

Fourier

|< < Prev Random Next > >|



That cat has some serious periodic components

|< < Prev Random Next > >|

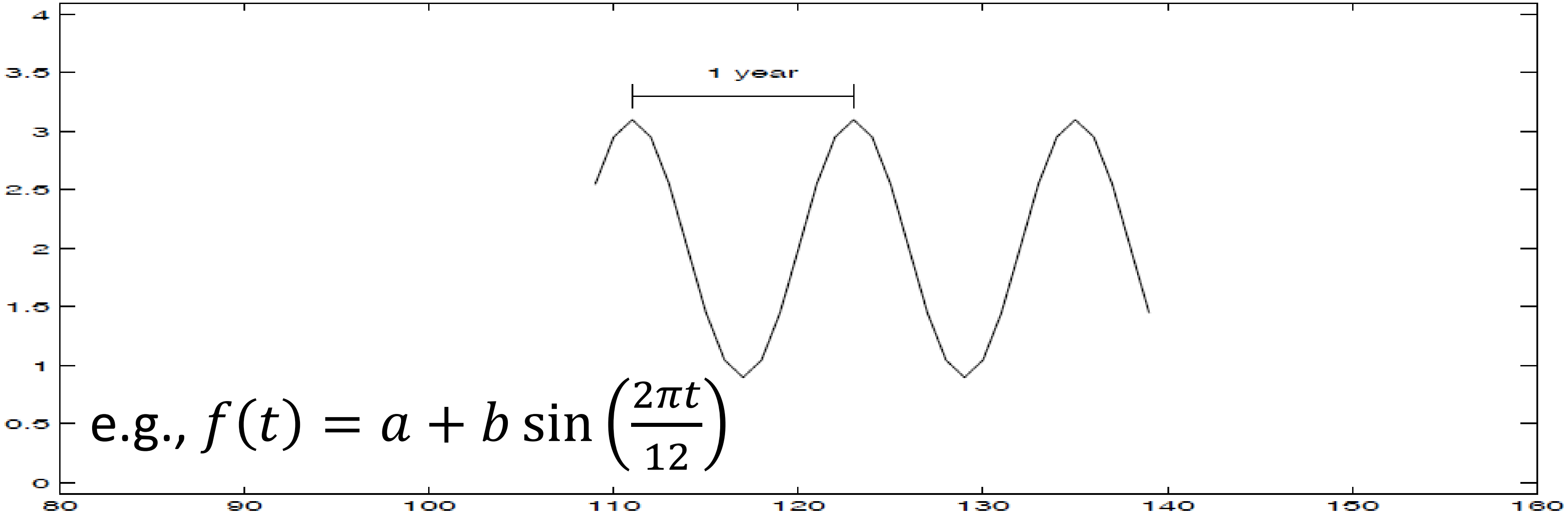
Permanent link to this comic: http://xkcd.com/26/
Image URL (for hotlinking/embedding): http://imgs.xkcd.com/comics/fourier.jpg

Example: Orange Juice Pricing

Example:

Consider the price of orange juice over many months.

Typically *cyclical* or *periodic* (i.e., pattern repeats over time) due to e.g. seasonal variation in supply and demand.

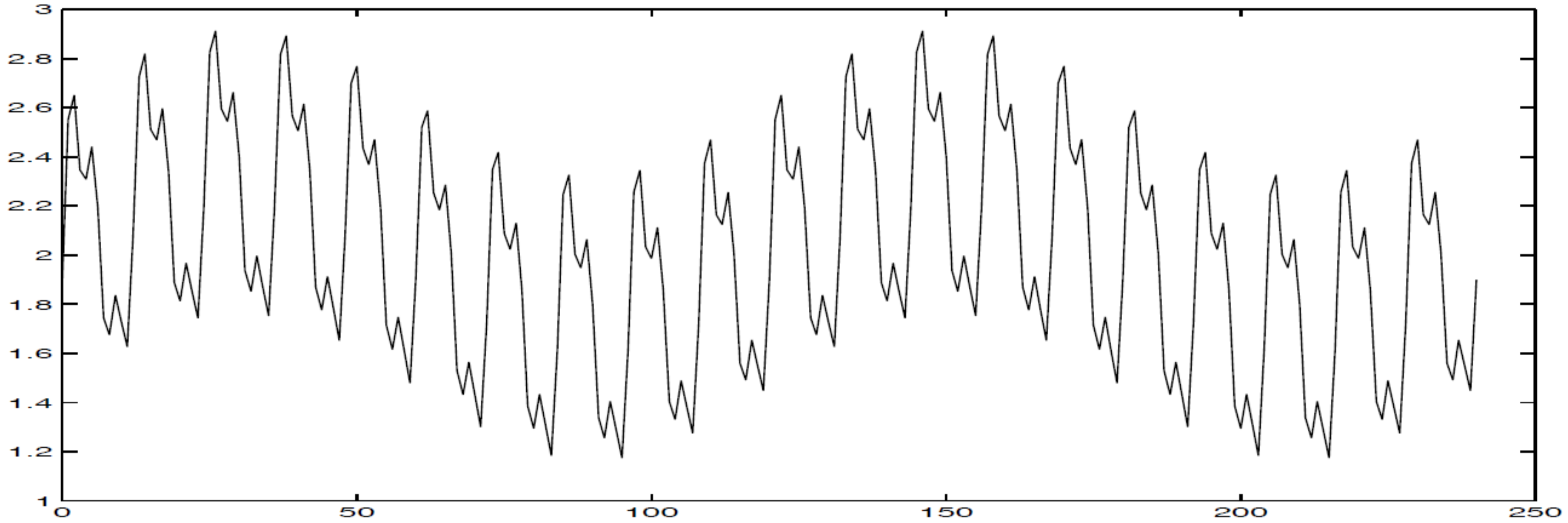


e.g., $f(t) = a + b \sin\left(\frac{2\pi t}{12}\right)$

Example: Orange Juice Pricing

Other recurring phenomena can have effects on different time scales: weather fluctuations, variation in import costs, El Nino, etc.

Data might actually look more like this:



Example: Orange Juice Pricing

How can we also represent these additional fluctuations in data?

Add more sinusoidal terms with *different* periods/frequencies.

General form could be:

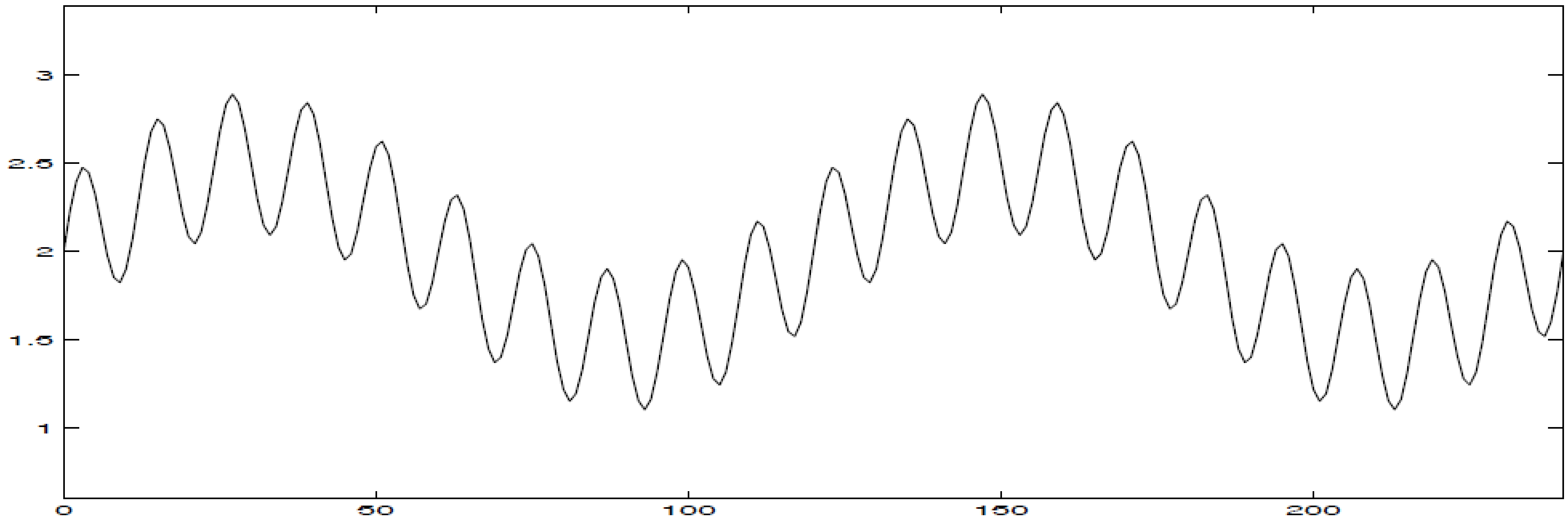
$$f(t) = a_0 + a_1 \cos(qt) + b_1 \sin(qt) + a_2 \cos(2qt) + b_2 \sin(2qt) + \cdots$$

In this case, $q = \frac{2\pi}{T}$ where T is the total time period, e.g. 240 months worth of pricing data.

Example: Orange Juice Pricing

e.g., Two sources/frequencies of fluctuations might give something like:

$$f(t) = a_0 + b_2 \sin(2q \cdot t) + b_{20} \sin(20q \cdot t)$$



Fourier Analysis – Basic Idea

Convert time or space-dependent data into the representation

$$f(t) = a_0 + a_1 \cos(qt) + b_1 \sin(qt) + a_2 \cos(2qt) + b_2 \sin(2qt) + \cdots$$

i.e. infinite sum of increasingly high frequency ***sine/cosines***.

The a_i and b_i coefficients now determine the function.

Could view this as:

- a weird (non-polynomial) interpolation problem: what coefficients let us fit this particular summation form to a given function?
- an expansion/approximation of the function as an infinite sum of sines/cosines. (e.g., compare with our dear friend, the Taylor series.)

Fourier Analysis – Main Topics

Main topics:

- How do we find the coefficients for this frequency-based representation?

- What if our input data are *discrete* values $f_0, f_1, f_2 \dots f_N$ instead of a smooth *continuous* function $f(t)$? (Discrete Fourier transform, DFT).

- How can we transform data into the frequency-domain *efficiently*? (The "**Fast** Fourier Transform", or FFT).

- Discussion of various applications. (i.e. what's the purpose?)

Continuous Fourier Series

Continuous Fourier Series

Consider some continuous ***periodic*** function $f(t)$ with period T, so

$$f(t \pm T) = f(t).$$

i.e., f repeats after one length/period T.

e.g., $\sin\left(\frac{2\pi kt}{T}\right)$ or $\cos\left(\frac{2\pi kt}{T}\right)$ satisfy this for all integers k.

$$\cos\left(\frac{2\pi k(t+T)}{T}\right) = \cos\left(\frac{2\pi kt}{T} + 2\pi k\right) = \cos\left(\frac{2\pi kt}{T}\right)$$

Continuous Fourier Series

Goal is to represent any $f(t)$ as an infinite sum of trig functions:

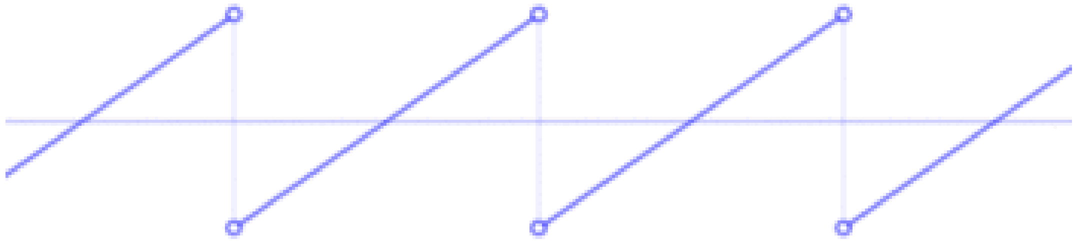
$$f(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos\left(\frac{2\pi kt}{T}\right) + \sum_{k=1}^{\infty} b_k \sin\left(\frac{2\pi kt}{T}\right)$$

a_k, b_k indicate the “information”/amplitude for each sinusoid of a specific period $\frac{T}{k}$, or frequency $\frac{k}{T}$.

Higher integer k indicates shorter period & higher wave frequency.

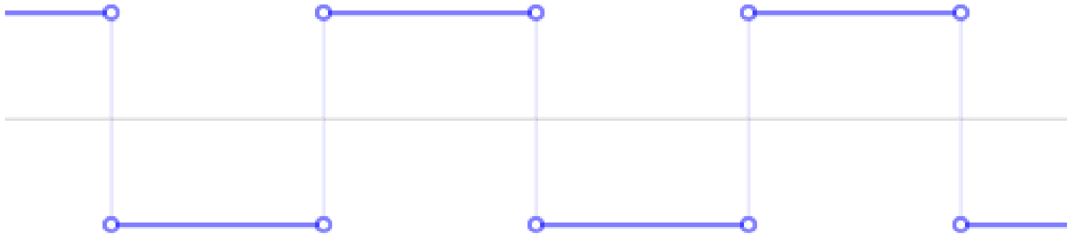
Examples –
More sinusoids give better approximations

“Sawtooth” wave:



N = 0

Square wave:



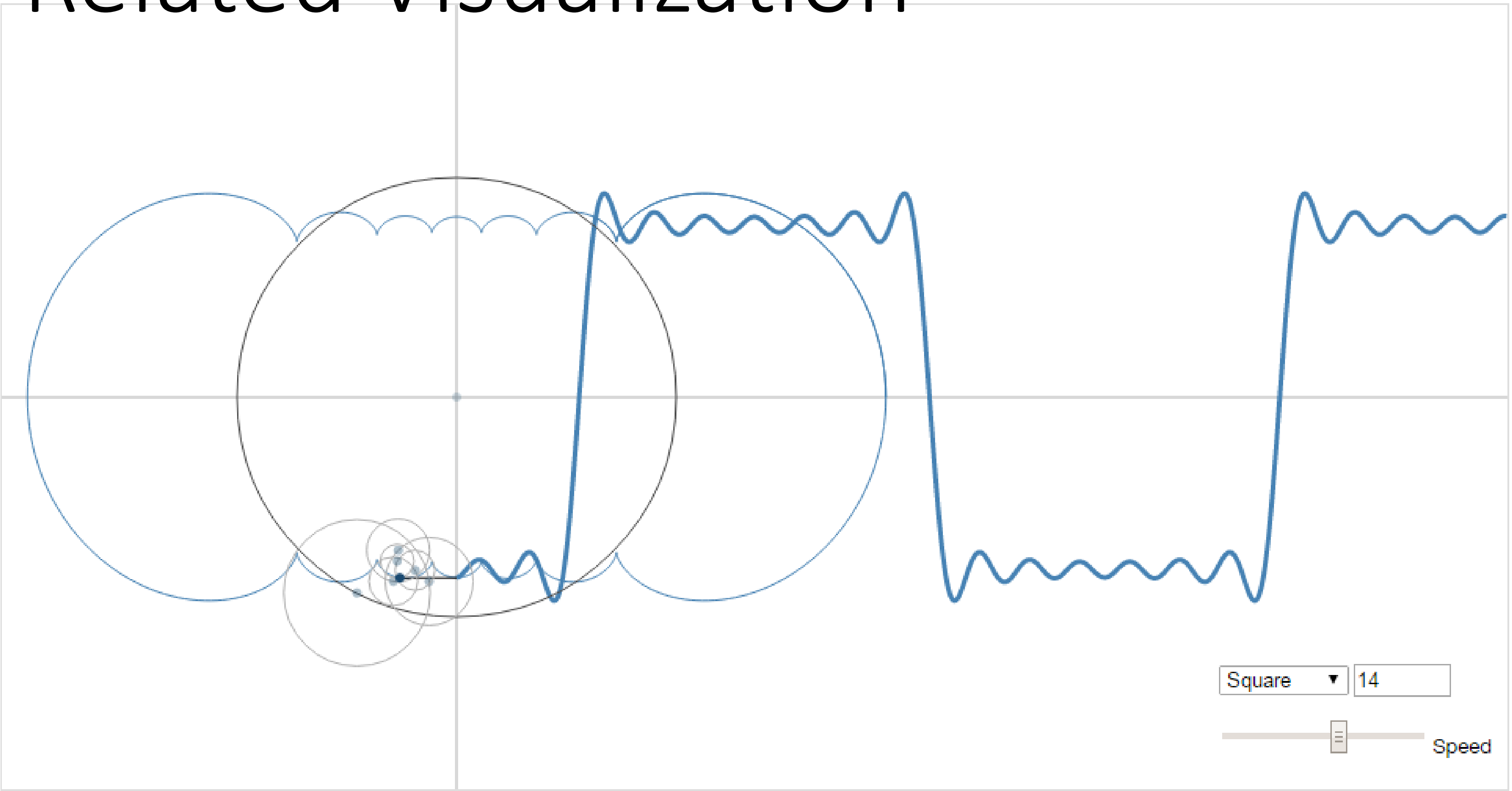
N = 0

[Images via LucasVB, Wikimedia]

A Snazzy Visualization –
Square wave as sum of sinusoids.

https://commons.wikimedia.org/wiki/File:Fourier_transform_time_and_frequency_domains.gif

A Related Visualization



From Wikipedia:

Open in a new window.

In mathematics, a **Fourier series** decomposes periodic functions or periodic signals into the sum of a (possibly infinite) set of simple oscillating functions, namely sines and cosines (or complex exponentials).

http://bl.ocks.org/jinroh/7524988

Determining the coefficients

Assume for now that range of t is $t \in [0,2\pi]$, and period $T = 2\pi$.

Then we have

$$f(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos(kt) + \sum_{k=1}^{\infty} b_k \sin(kt)$$

We will now consider some handy identities that let us determine each coefficient **alone**.

Handy Identities (Orthogonality)

First handy identity:

$$\int_0^{2\pi} \cos(kt)\sin(k't)\,dt = 0 \text{ for any integers } k, k'.$$

i.e., the integral of the product of $\cos(kt)$ and $\sin(k't)$ over on $[0,2\pi]$ is 0.

We say that these two functions are **orthogonal** to each other.

You can try verifying this using trig identities (e.g., product-to-sum formulas).

More Orthogonality Relations

$$\int_0^{2\pi} \cos(kt)\cos(k't)\,dt = 0 \text{ for } k \neq k'$$

$$\int_0^{2\pi} \sin(kt)\sin(k't)\,dt = 0 \text{ for } k \neq k'$$

$$\int_0^{2\pi} \sin(kt)\,dt = 0$$

$$\int_0^{2\pi} \cos(kt)\,dt = 0$$

We can use these to extract a single Fourier coefficient at a time!

Let's see how (by hand).

To the board we go!

Determining the coefficients

So we can find all the coefficients by solving integrals:

$$a_0 = \frac{\int_0^{2\pi} f(t)dt}{2\pi}$$

$$a_k = \frac{\int_0^{2\pi} f(t)\cos(kt)dt}{\int_0^{2\pi} \cos^2(kt)\,dt}$$

$$b_k = \frac{\int_0^{2\pi} f(t)\sin(kt)dt}{\int_0^{2\pi} \sin^2(kt)\,dt}$$

How does *phase* of the sinusoids come in?

What if we have same function $f(t)$, but with a different phase? (ie., shifted in time/space.)

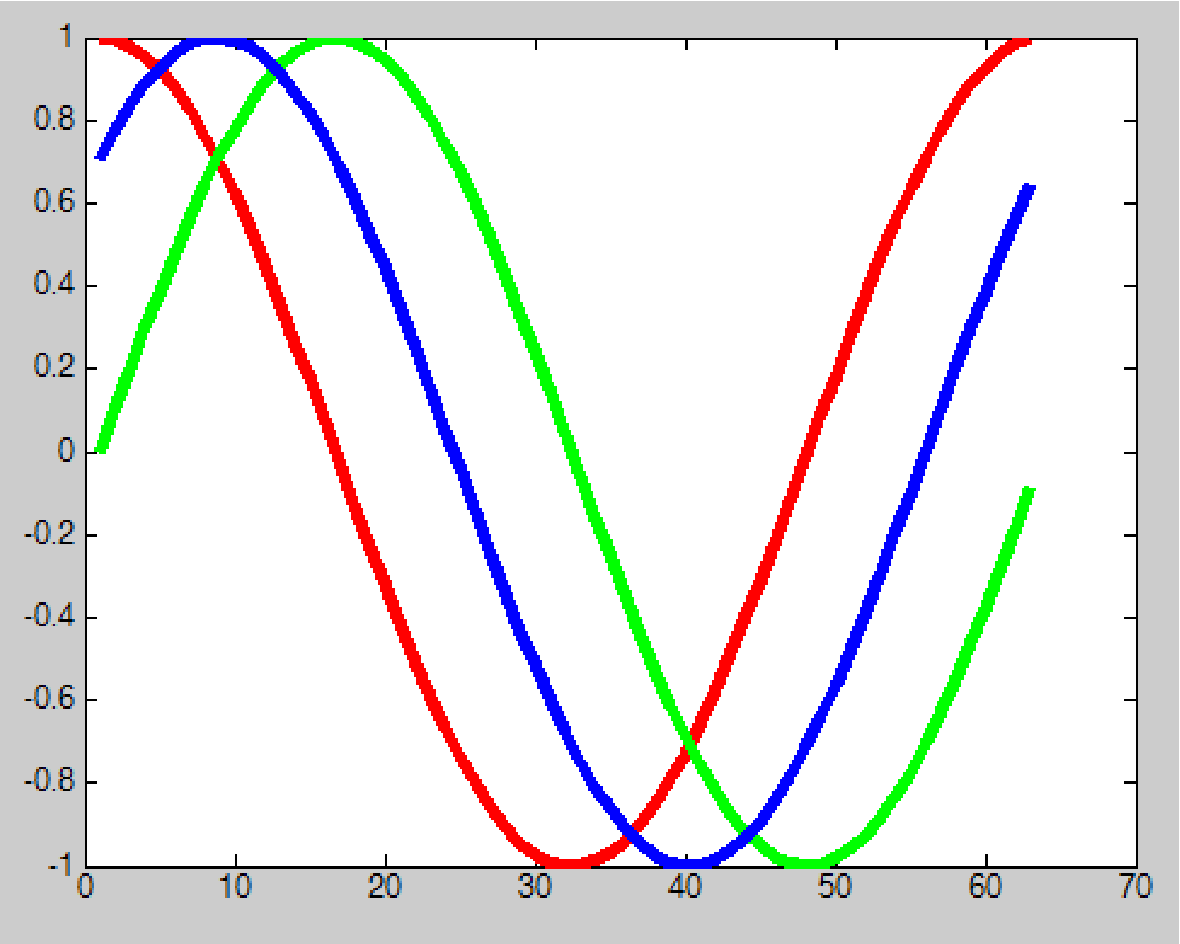
A sine is a translated cosine! All "in-between" shifts are just *linear combinations* of sin/cos.

Red = $\cos(x)$

Green = $\sin(x)$

Blue = $\frac{1}{\sqrt{(2)}}\cos(x) + \frac{1}{\sqrt{(2)}}\sin(x)$

Changing the phase of the input function "transfers" some weight between a_k and b_k coefficients.



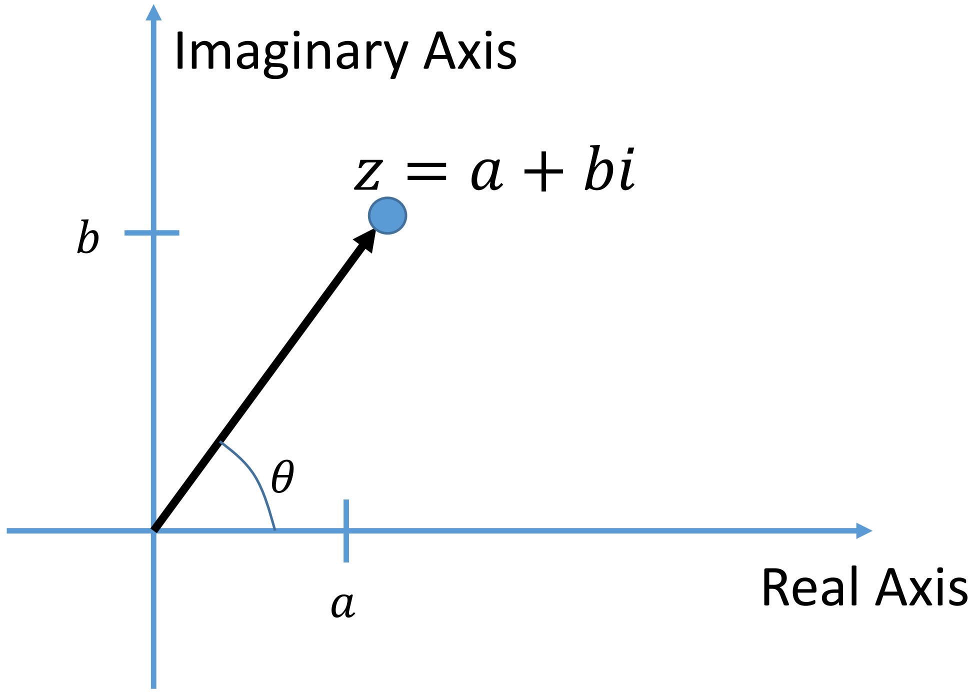
Complex Number Review

We will make extensive use of complex numbers C from here on.

For $z \in C$, we write $z = a + bi$ where the imaginary number $i = \sqrt{-1}$.

Visualized as points on the ***complex plane***:

The vector length and angle are often called "modulus" and "argument", respectively.



Complex Number Review

Various operations such as addition, multiplication, etc. are all defined for complex numbers.

Just need to remember that $i^2 = -1$.

For $z = a + bi$, we can define additional operations:

- Conjugate: $\bar{z} = a - bi$
- Modulus/norm: $|z| = \sqrt{a^2 + b^2}$
- Argument/phase: $\mathrm{Arg}(z) = atan2(b, a)$

Complex Numbers - Examples

Addition: $(1+2i)-(3-2i)=-2+4i$

Multiplication: $(1+2i)\cdot(3-2i)=3+6i-2i-4i^2$
$$=3+4i-(-4)=7+4i$$

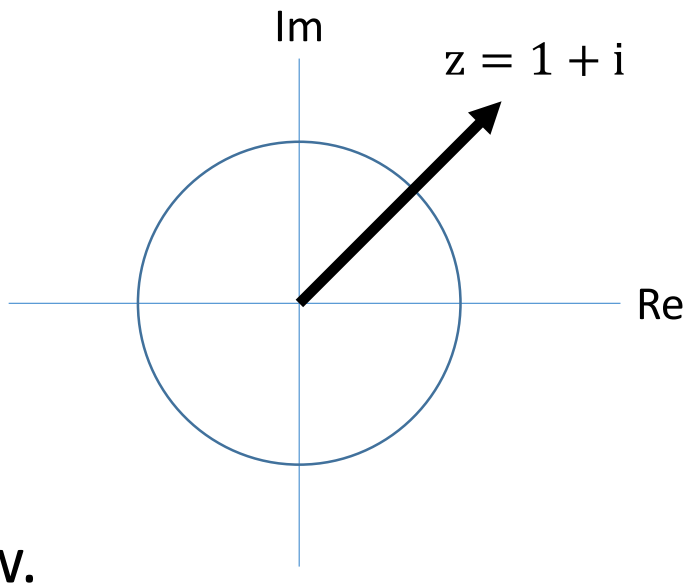
Conjugate: $\overline{3-2i}=3+2i$

Modulus: $|3-2i|=\sqrt{3^2+2^2}=\sqrt{13}$

Argument: $\text{Arg}(1+i)=\pi/4.$

Matlab has extensive support for complex numbers.

See **Appendix E** in the course notes for additional review.



Using Euler’s formula

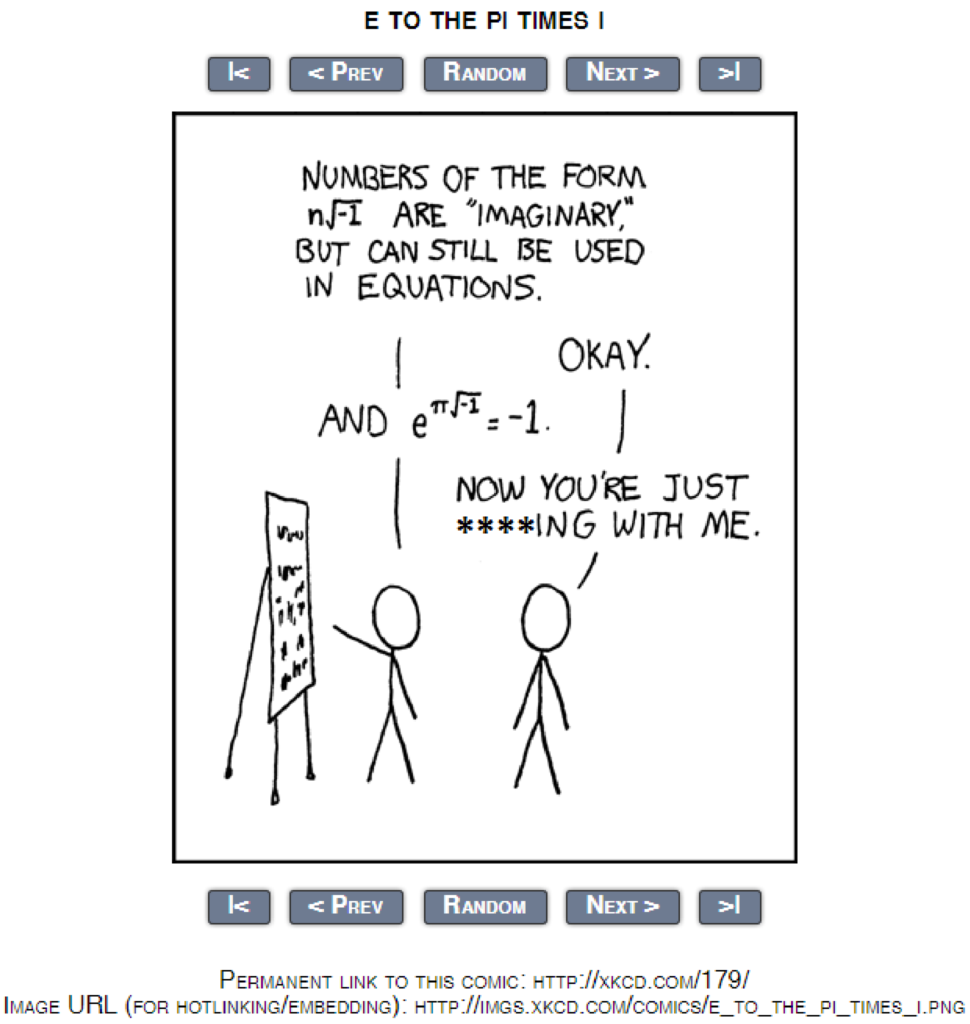
We will find ***Euler’s formula*** very useful:

$$e^{i\theta} = \cos(\theta) + i\sin(\theta)\,.$$

Also $e^{-i\theta} = \cos(\theta) - i\sin(\theta)\,.$

Adding/subtracting these lead to two key identities:

$$\cos(\theta) = \frac{e^{i\theta}+e^{-i\theta}}{2} \text{ and } \sin(\theta) = \frac{e^{i\theta}-e^{-i\theta}}{2i}.$$



Fourier series with complex exponentials

Now, given our earlier sinusoidal expression of a function $f(t)$

$$f(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos(kt) + \sum_{k=1}^{\infty} b_k \sin(kt)\,,$$

we can express it more concisely as

$$f(t) = \sum_{k=-\infty}^{+\infty} c_k e^{ikt}$$

where the c_k coefficients are **complex numbers**.

There exists a simple conversion: $c_k \leftrightarrow\ a_k, b_k$.

Converting between c_k, and a_k and b_k

Specifically, for $k > 0$:

$$a_0 = c_0$$

$$c_k = \frac{a_k}{2} - \frac{ib_k}{2}$$

$$c_{-k} = \frac{a_k}{2} + \frac{ib_k}{2}$$

(These likewise fall out from Euler’s formula.)

Converting between c_k, and a_k and b_k

And we have the relationships:

$$|a_0| = |c_0| \text{ and } |c_k| = |c_{-k}| = \frac{1}{2}\sqrt{a_k^2 + b_k^2}$$

Modulus of c_k gives the amplitude/magnitude of a given frequency of waves.

Angle/argument of c_k gives that frequency's phase: $\theta = \mathrm{Arg}(c_k)$

Finding c_k coefficients

We can also find c_k directly, similar to what we did for a_k and b_k.

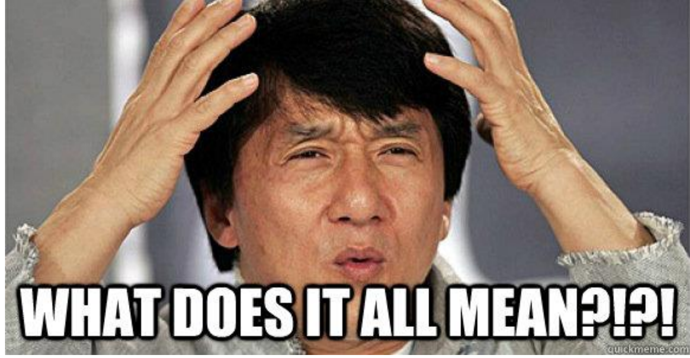
The necessary corresponding *orthogonality* property is…

$$\int_0^{2\pi} e^{ikt}e^{-ilt}dt = \begin{cases} 0; & k \neq l \\ 2\pi; & k = l \end{cases}$$

from which we can similarly find the coefficient formula

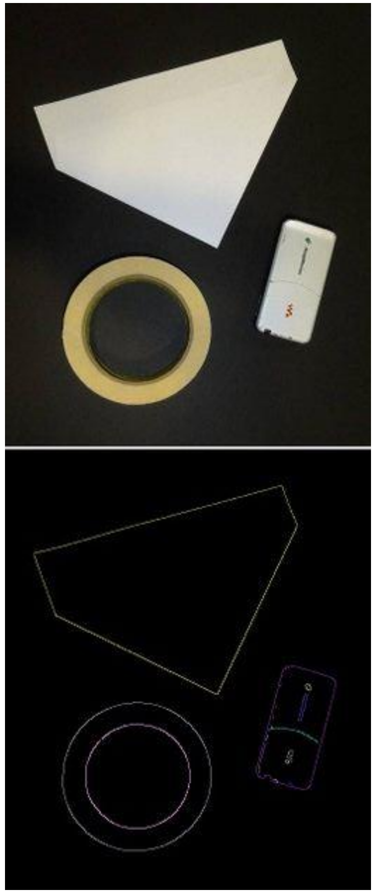
$$c_k = \frac{1}{2\pi}\int_0^{2\pi} e^{-ikt}f(t)dt\,.$$

Extracting Meaning



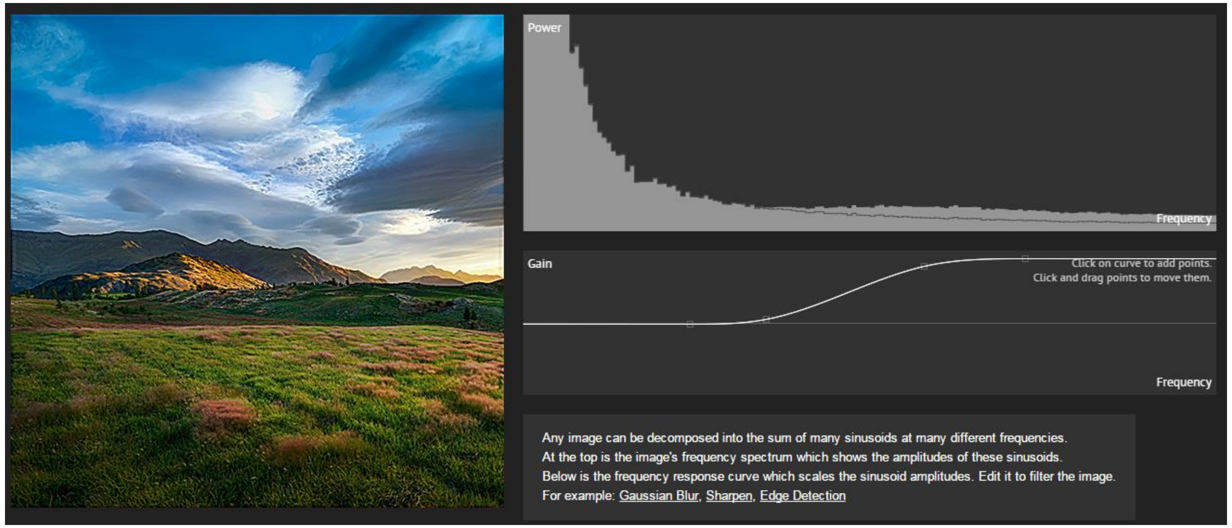
But what does it all *mean?* Various "physical" interpretations, depending on context:

- In electrical signals, the $\left|c_j\right|$ describe the *power* at given frequency $\frac{k}{T}$.
- High frequencies (i.e., large k terms) are often *noise* in a signal. *Filtering* out (dropping) these frequencies may clean the data.
- Or, high frequency image components might suggest *edges (discontinuities)*. Can be used to detect features, or sharpen/enhance edges.

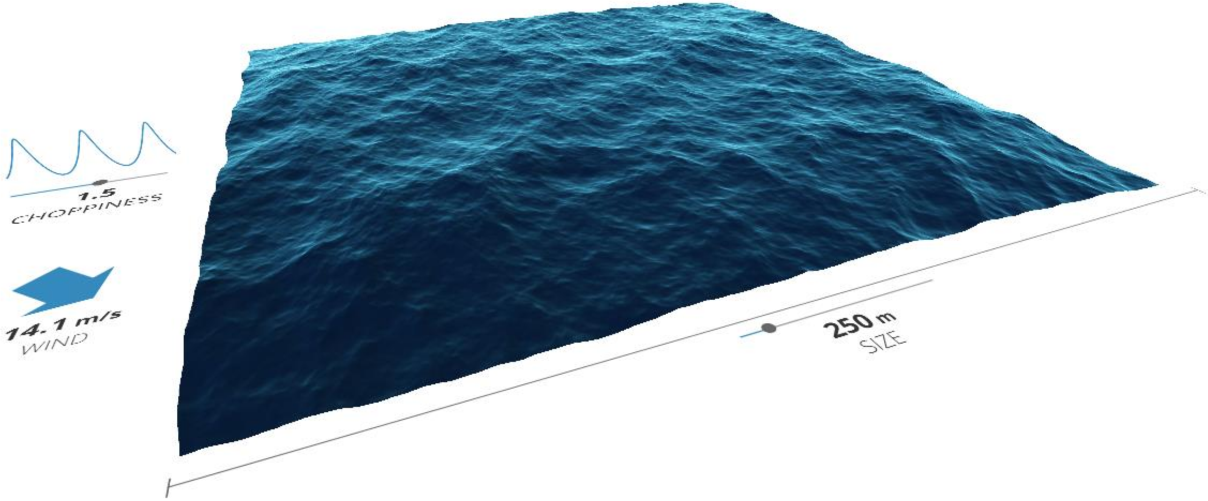


Edge Detection

Two FT Demo Applications – by David.Li



Fourier Image Editing: http://david.li/filtering/



FFT-Based Wave Simulation: http://david.li/waves/

Summary

- The Fourier transform converts functions/data into a sum of sinusoids terms which reveals its underlying ***frequency information***.
- This representation can be useful in many applications that manipulate various kinds of data.

An in-depth, interactive, and intuitive signal processing tutorial that touches on many concepts from this unit:

http://jackschaedler.github.io/circles-sines-signals/

Next Time

- Begin looking at the *Discrete Fourier Transform (DFT)*.
- It is the analog of the continuous Fourier transform applied to a vector of **discrete** data values, $f_1, f_2, f_3, \ldots, f_n$.