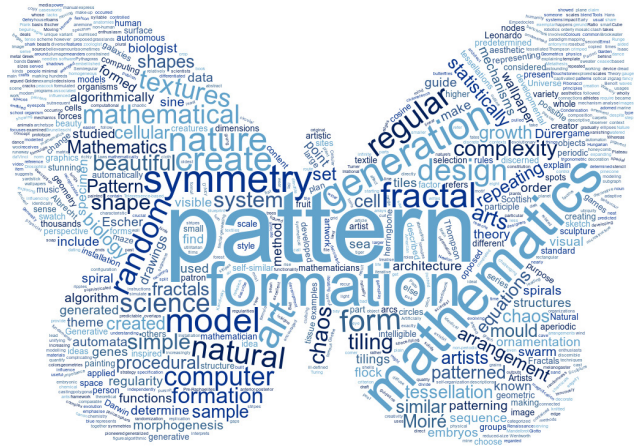
Advanced Mathematical Perspectives 1

Lecture 8: Sinusoids and the Fast Fourier Transform



Matthew Roughan

<matthew.roughan@adelaide.edu.au>

www.maths.adelaide.edu.au/matthew.roughan/notes/AMP1/

School of Mathematical Sciences,
University of Adelaide

THE UNIVERSITY of ADELAIDE

SUB CRUCE LUMEN

ACEMS

AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE FOR MATHEMATICAL AND STATISTICAL FRONTIERS

Section 1

Fourier transforms

Fourier's Theorem is not only one of the most beautiful results of modern analysis, but it is said to furnish an indispensable instrument in the treatment of nearly every recondite question in modern physics.

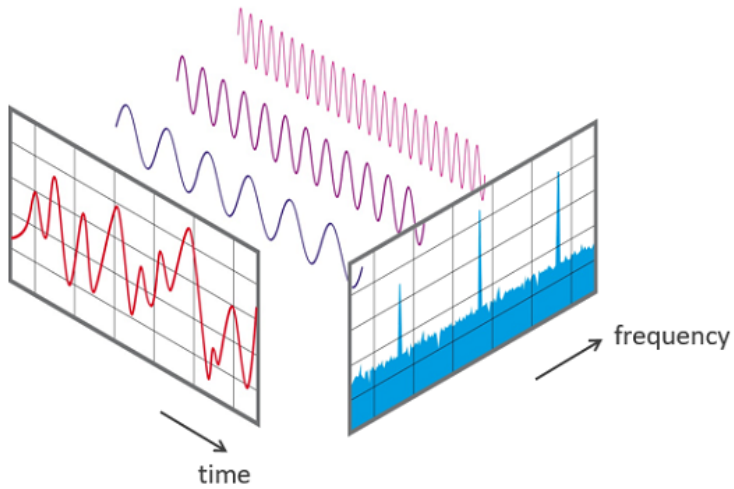
Lord Kelvin

Question

- I said before that "We can model *any* periodic function as a (possibly infinite) sum of sinusoids"
- But how would we do that?

Fast Fourier Transform

- The Fourier Transform transforms a signal from the *time domain* into the *frequency domain*
  - But note, it works with *complex* sinusoids
  - What I plotted above was the *power spectrum*, which is the squared magnitude of the Fourier transform



https://en.wikipedia.org/wiki/File:FFT-Time-Frequency-View.png

- The *Fast* Fourier Transform (FFT) is an *algorithm* for doing this *really* fast on real data

Complex Sinusoids

$$x = a + ib, \text{ where } i = \sqrt{-1}$$

- real part of x is $\Re(x) = a$
- imaginary part of x is $\Im(x) = b$
- complex conjugate $x^* = a - ib$
- Hermitian of a complex matrix $A = [a_{ij}]$ is $A^H = [a^*_{ji}]$.
- identities
  - $e^{ix} = \cos(x) + i\sin(x)$
  - $\cos(x) = \frac{1}{2}\left(e^{ix} + e^{-ix}\right)$
  - $\sin(x) = \frac{1}{2i}\left(e^{ix} - e^{-ix}\right)$

FFT in practice

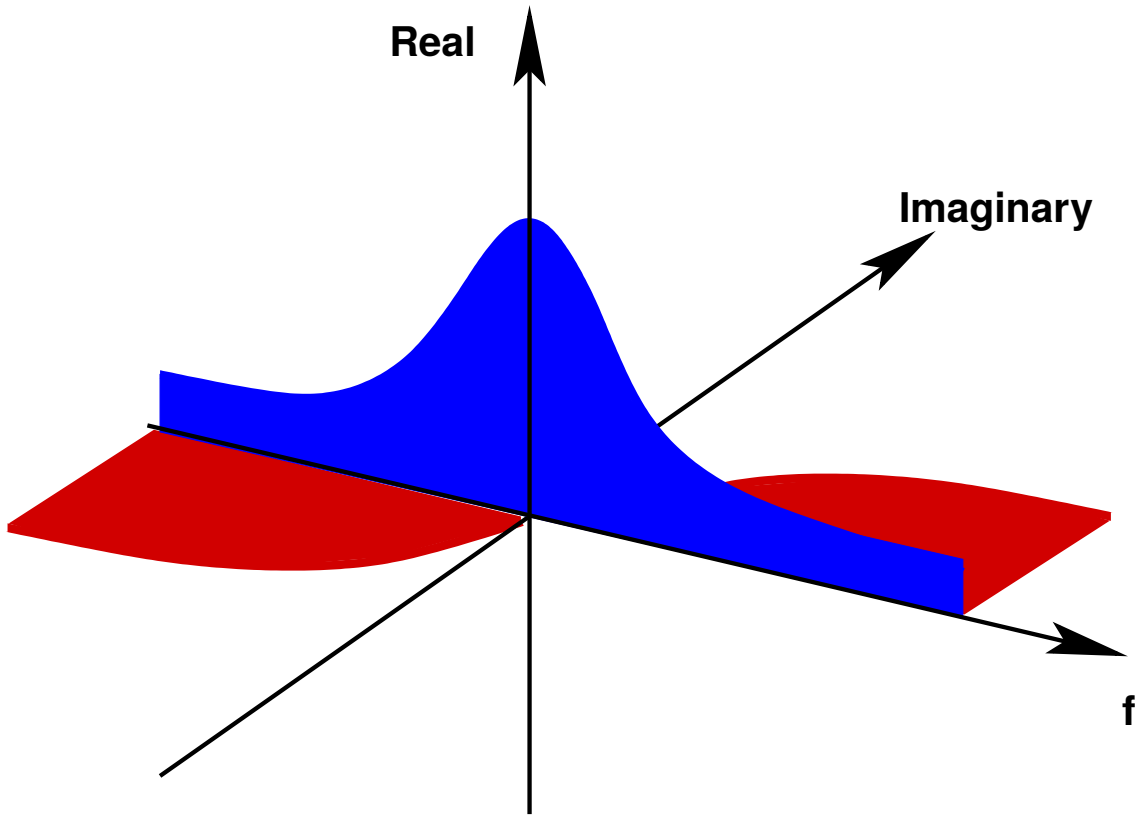
- I don't have time to tell you all I would like about FFTs
- But MATLAB has several functions that will do FFTs for you

```
fft
fft2
fftshift
ifft
```

https://au.mathworks.com/help/matlab/ref/fft.html
https://au.mathworks.com/help/matlab/math/fourier-transforms.html
https://au.mathworks.com/help/matlab/examples/using-fft.html
https://au.mathworks.com/help/matlab/math/two-dimensional-fft.html

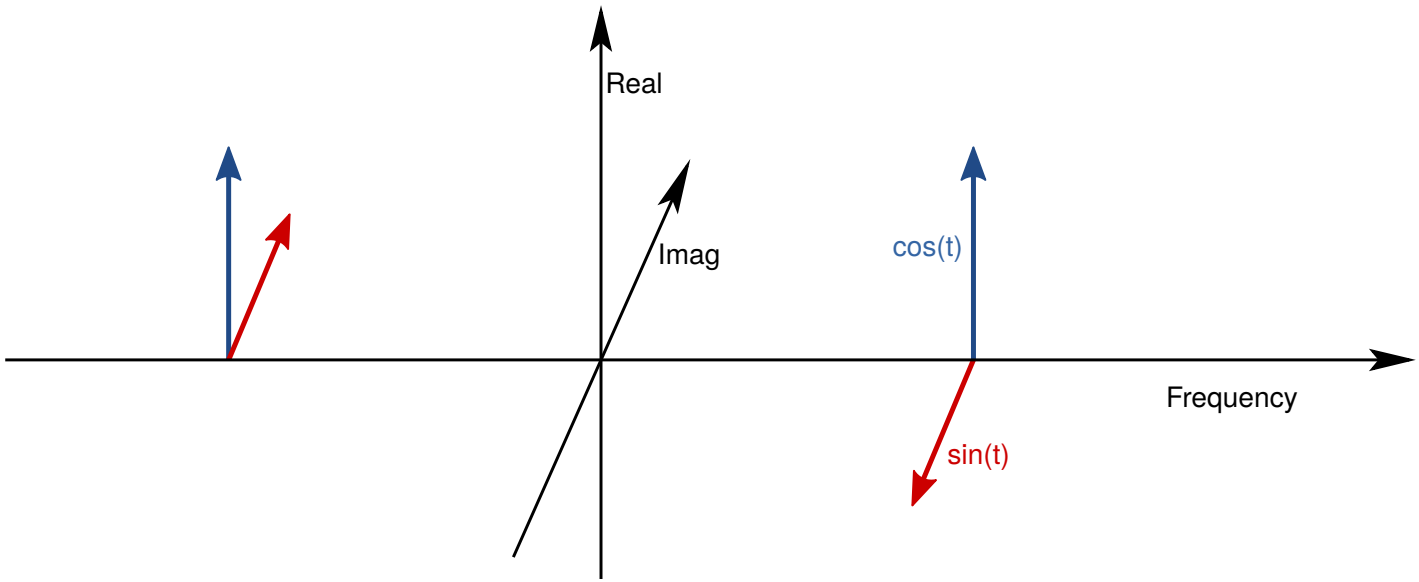
Things to note

- If you start with a real-value signal (which we almost always do), then the Fourier Transform will be Hermitian symmetric $y(-f) = y^*(f)$
  - real part is symmetric
  - complex part is anti-symmetric

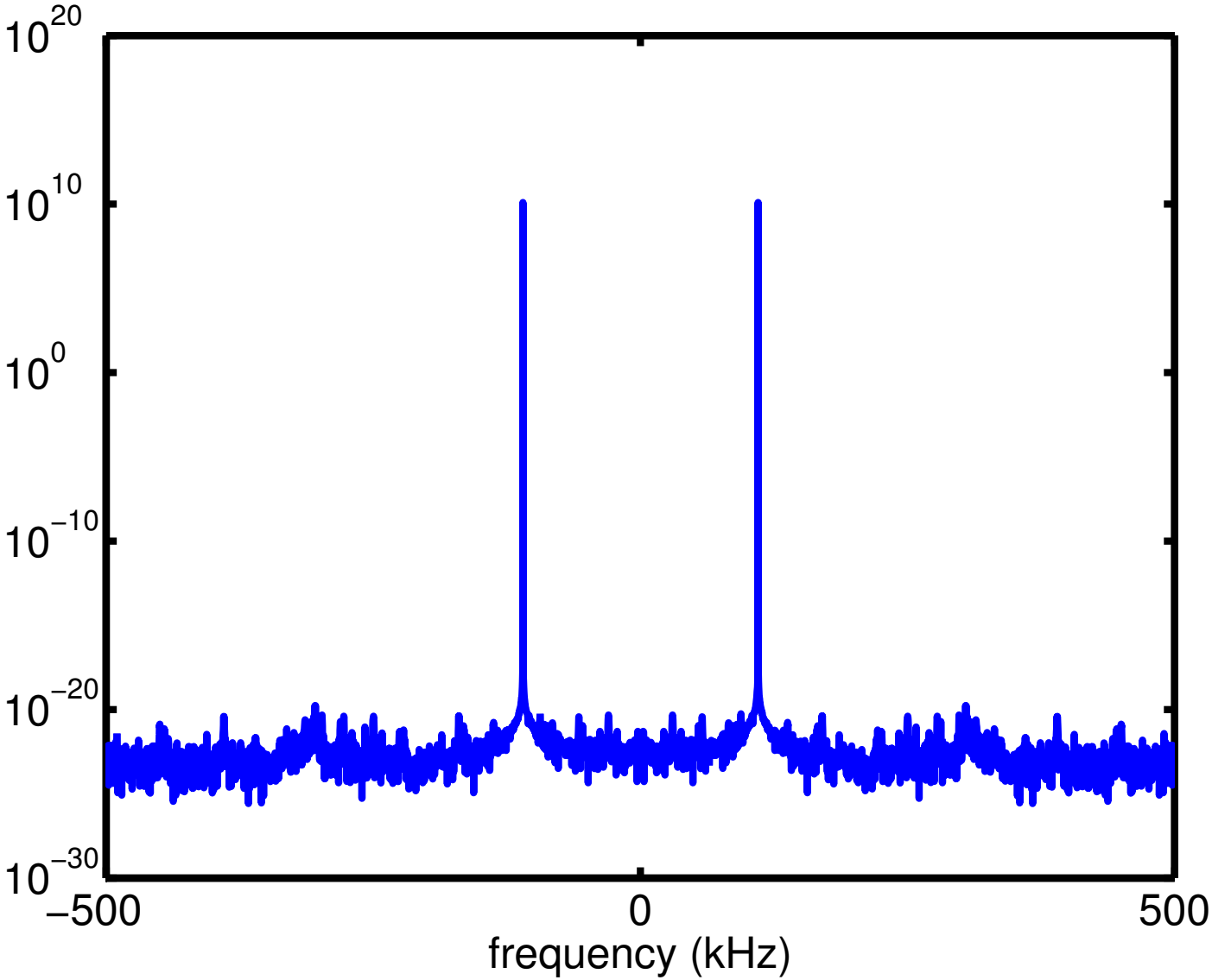


- Real FFTs will have *noise*

Simple examples of Fourier Transforms



Real FFT of a cosine



Activity

Perform FFTs on some examples.

Further reading I