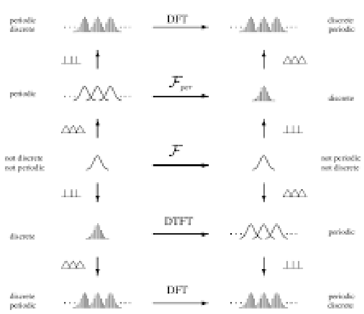
What is Fast Fourier Transform?
Muireann Heaney
M105 Mathematical Analysis 2 Final Essay

I was first introduced to the idea of a Fourier Transform (FT) in my Fourier Methods course in sophomore year. Then, I was completely unaware of the uses of Fourier Analysis and completely confused on why I needed an entire course to learn about Fourier Series, Fourier Transforms, Residues and so on. I'm studying Math, Applied Math and Physics and since then I have seen this class was probably one of the most important ones as it has been prominent in every class since and in some cases the saving grace! Such as, electromagnetism, analytic mechanics, quantum mechanics and my physics and music class. In each class I learned different uses of these methods, so when we started this in M105 I was excited to learn another perspective which is definitely completely different to what I had experienced before. I'm going to first introduce the idea of a FT and then a FFT (Fast Fourier Transform).



Firstly, what is a Fourier Transform(FT)?
A FT is a function(transformation) that can break down another function into it's components. The one I find easiest to visualize is the decomposition of signals into their different frequencies, or a sound wave broken up into it's frequencies. If you hear a sound (musical wave) and want to know how to reproduce it, you must know it's components so what do you do? Use a Fourier Transform to get the pitches present. To summarize, any study of waves I have come across it is natural that a FT pops up. Fourier Methods are usually applied to complex-valued functions and maybe vector functions.

As we saw in class, (Pugh Chapter 5) The FT can be defined as an improper Riemann integral.

$$ft(n) := < f, e_n > = \int_{[0,1]} f(x)e^{-2\pi i n x} dx$$

$$The\ function\ ft: Z \rightarrow C\ is\ called\ the\ Fourier\ transform\ of\ f$$

It can send functions of different spaces from one to another e.g *3D position space to 3D momentum space.* For example in quantum mechanics it is useful to be able to represent a waveform as a function. Now, looking from a pure math perspective on how fourier methods can be generalized on groups, which includes the discrete-time Fourier Transform (DTFT, group=Z), the discrete Fourier transform (DFT, group= Z modN) and the Fourier Series (Wiki).

What is a FFT?
In short, the FFT is an algorithm that determines the DFT of an input significantly *faster* than computing it directly. So what is a DFT?

A DFT is a method of converting a sequence of complex numbers into a new sequence of complex numbers.
Taking a sequence of N complex numbers:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi ikn/N} \; for\; 0 \leq k \leq N-1$$

x_i are the values of the input function at equally spaced times $t = 0$,...., $N - 1$. So when we input a sequence of complex numbers $\{x_i\}$ we get another sequence of complex numbers $\{X_k\}$. The output represents the amplitude and phase of a sinusoidal wave with frequency k/N(brilliant.org). The point of computing the DFT is to approximate a signal by a linear combination of waves. There are lots of applications and they are involved in multiplying large polynomials together which can take some time and if by hand (rare I know) can have a larger percentage error. But not to worry, the DFT can be computed by the Fast Fourier Transform.

Finally, lets define an FFT!
The fast Fourier transform (FFT) is a discrete Fourier transform algorithm which reduces the number of computations needed for N points from $2N^2$ *to* $2Nlog_2N$ *(Wolfram).*
FFT's fall into two classes: composing in time, or in frequency. An FFT computes transformations by factorizing the DFT matrix into a product of sparse factors. This has a huge effect on time needed to calculate DFT's, especially seen in coding with large data sets. There are many algorithms for complex-number arithmetic, group theory and number theory(Wiki).

As we can see FFT's are very useful in many areas and can speed up a lot of computations. Many FFT algorithms depend only on the fact that $e^{-2\pi i/N}$ is an N-th primitive root of unity (Any complex number that yields 1 when raised to some positive integer n), and thus can be applied to analogous transforms over any finite field, such as number-theoretic transforms *(NTT- A finite field and primitive nth roots of unity exist whenever n divides p-1).*

As you can see, FFT's are very useful and we have all most likely come across them already, or will come across them in the future.