# The Fourier Transform: Variations and Applications

Sailesh Kaveti

## 1. The Fourier Transform

### 1.1. Developing an Intuition for the Fourier Transform.

Imagine we are given we are given a bucket of paint, and asked to replicate the color of that bucket. At first, it seems easy to simply describe it nominally, calling the bucket of paint as red, yellow, blue, or any other color. This is a valid first instinct. However, imagine you were given a purple bucket of paint. Even in this instance, most people are familiar with the color purple and we could just mix equal parts of blue and red. Imagine you were given a bucket of paint that was dark purple. We could still call this bucket "dark purple", but we would immediately run into problems when replicating the paint color. Do we need to put the colors on a 2:1 ratio or a 3:1 ratio? In general, given a color, it can be very useful to know the colors that compose that color, as opposed to just a nominal description. Now imagine that we were given a complex wave, and that we are interested in the simpler waves that compose that wave. The Fourier Transform helps us determine a periodic wave's composite frequencies.

### 1.2. The Fourier Series.

In order to understand the Fourier Transform, it is very important to gain understand the Fourier Series and its integral and the Fourier Transform is non-discretized extension of the Fourier Series. According to Dym and McKean, the authors of *Fourier Series and Integrals*, the basic idea of the Fourier Series is that any periodic function $f(t)$ can be expressed as a trigonometric sum of sines and cosines. More specifically, for the same period T for both *sine* and *cosine*, we can claim the following:

$$f(t) = \sum_{n=0}^{\infty} [\hat{f}_+(n) \cos\left(\frac{\pi n t}{T}\right) + \hat{f}_-(n) \sin\left(\frac{\pi n t}{T}\right)]$$

From this, we can derive the equations for $\hat{f}_+^a(n)$ and $\hat{f}_-^b(n)$. The equations are as follows

$$\hat{f}_+(n) = \frac{1}{L} \int_{-L}^{L} f(t) cos\Big(\frac{n\pi x}{L}\Big) dx$$

$$\hat{f}_-(n) = \frac{1}{L} \int_{-L}^{L} f(t) sin\Big(\frac{n\pi x}{L}\Big) dx$$

Putting these two together, we can get the Fourier Series, a formula to completely put a periodic function in terms of *sine* and *cosine*. For example, a non-analytic square wave $g(x)$, that has a value of $-h$ from $-\pi$ to 0 and h from 0 to $pi$ with a vertical line from $-h$ to $h$ can be modeled with the following Fourier Series:

$$g(x) = \sum_{n=0}^{\infty} \frac{sin((2n+1)x)}{(2n+1)}$$

The intuition behind a Fourier Series is very similar to a Taylor or MacLaurin Series, where a complex function can be modeled and very closely as an infinite sum of simpler functions.

### 1.3. The Computation of the Fourier Transform.

Similar to colors, given a complex wave, it is important to know the frequencies that compose that wave. The inverse of this can also provide us with valuable information. A wave $f(x)$ exists in real space, and its equivalent in frequency space is function $F(s)$. In short, $f(x)$ gives us a complex wave in real space, while $F(s)$ gives us a function with peaks at the frequencies that make up $f(x)$.

As described in *The Fourier Transforms and Its Applications*, the following relationship between F(s) and f(x) exists:

The Fourier Transform:

$$F(s) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi xs}dx$$

The Inverse Fourier Transform:

$$f(x) = \int_{-\infty}^{\infty} F(s)e^{i2\pi xs}ds$$

Replacing any instances of $e^{i\theta}$ with $cos(\theta)+isin(\theta)$, we can derive the following equivalent expressions:

The Fourier Transform:

$$F(s) = \int_{-\infty}^{\infty} f(x)(cos(-2\pi xs) + isin(-2\pi xs))dx$$

The Inverse Fourier Transform:

$$f(x) = \int_{-\infty}^{\infty} F(s)(cos(2\pi xs) + isin(2\pi xs))ds$$

As a result, we can see a resemblance of a non-discretized version of the Fourier Series, which we can discretize for more interesting results in the following section.

## 2. The Discrete Fourier Transform

### 2.1. The Basics of the Discrete Fourier Transform.

### 2.2. How the Discrete Fourier Transform varies from the Fourier Transform.

## 3. The Computation of the Discrete Fourier Transform

### 3.1. A Rudimentary Algorithm for the Discrete Fourier Transform.

### 3.2. The Fast Fourier Tranform.

## 4. Applications of the Fourier Transform

### 4.1. Convolution of Polynomials.

According the *Algorithms*, the straightforward method of adding two polynomials takes $O(n)$ time. We know the following to hold true:

$$f(x) = \sum_{j=0}^{n} a_j x^j \tag{1}$$

$$g(x) = \sum_{j=0}^{n} b_j x^j \tag{2}$$

$$f(x) + g(x) = \sum_{j=0}^{n} (a_j + b_j) x^j \tag{3}$$

For two polynomials $f(x)$ and $g(x)$, with length $k$ and $n$, we know that $f(x) + g(x)$ has length $max(k, n)$. Generate two arrays of length $n$ called $A_f$ and $A_g$ where each index $i$ represents the coefficient for term $x^i$ for $f(x)$ and $g(x)$ respectively. Iterate through the arrays and add the value held at each index of $A_f[i]$ and $A_g[i]$ and set that value in $A_{f+g}[i]$. The resulting array represents the coefficients of $f(x) + g(x)$.

With this in mind, we can derive the straightforward algorithm for the multiplication of two polynomials. For the multiplication of the two polynomials, Cormen claims that the following holds true:

$$f(x) = \sum_{j=0}^{n} a_j x^j \tag{4}$$

$$g(x) = \sum_{j=0}^{n} b_j x^j \tag{5}$$

$$f(x)g(x) = \sum_{j=0}^{2n} c_j x^j \tag{6}$$

where

$$c_j = \sum_{k=0}^{j} a_k b_{j-k} \tag{7}$$

From this information, we can see that equation (6) yields us with the sum of $2n$ polynomials, each of which was computed in $O(n)$ time. As we showed earlier, the sum of any two polynomials can be computed in $O(n)$ time, and since we must add $2n$ total polynomials, we get that the product of any two polynomials can be computed in $O(2n^2)$ time, which can be reduced to $O(n^2)$ time by the definition of big-$O$ notation.

With our understanding of the FFT, Cormen claims that we can find the product of any two polynomials in $O(nlog(n))$ time, a significant asymptotic improvement over $O(n^2)$.

**4.2. Image Processing.**