The Fourier Transform

By adding a number of sine waves of different frequencies and amplitudes, we can approximate any given periodic function. However, if we limit ourselves to only sine waves with no offset, i.e., difference in phase at \( x = 0 \), we cannot perfectly reconstruct every function. For instance, the output of our function at \( x = 0 \) must always be 0. We could add a phase to each sine wave, but as Monsieur Fourier found out, it is sufficient to use both sine and cosine waves (which describe the same wave, but with a 90° phase difference) and keep all phases at 0.

The Fourier Transform

So for any given function \( A \), we could define a function \( \hat{A}_s \) that assigns to any frequency an amplitude (i.e., weight) that the sine wave of that frequency will receive in our sum of sine waves. Similarly, we could define a function \( \hat{A}_c \) that does the same for the cosine waves. In other words, these functions describe the contribution of different frequencies to the overall signal, our function \( A \).

While \( A \) describes our function in the spatial domain (i.e., what value does it have in what position \( x \)), \( \hat{A}_s \) and \( \hat{A}_c \) describe the same function in the frequency domain (i.e., the extent to which different frequencies are included in the function).

The Discrete Fourier Transform (DFT)

If we sample a function in constant spatial or temporal steps, we can perform a discrete Fourier transform on the resulting discrete data. A nice property of the DFT is that if we take \( n \) samples per period, we only need \( n \) sine and \( n \) cosine coefficients to perfectly represent the signal. The inverse DFT can then reconstruct the original signal without any loss.

\[
A[j] = \sum_{l=0}^{n-1} \left[ \hat{A}_s[l] \cdot \cos \left( 2 \pi \frac{j \cdot l}{n} \right) - \hat{A}_c[l] \cdot \sin \left( 2 \pi \frac{j \cdot l}{n} \right) \right]
\]

Here, the frequency \( l \) ranges from 0 to (\( n - 1 \)), representing the number of periods of a sinusoid (sine or cosine) that fit into the sampled interval of \( A \).
The Discrete Fourier Transform (DFT)

We can now finally write the equations for the DFT:

\[
\hat{A}_c[l] = \frac{1}{n} \sum_{j=0}^{n-1} A[j] \cos \left(\frac{2\pi \cdot lj}{n}\right)
\]

\[
\hat{A}_s[l] = -\frac{1}{n} \sum_{j=0}^{n-1} A[j] \sin \left(\frac{2\pi \cdot lj}{n}\right)
\]

Refresher: Complex Numbers

As you certainly remember, complex numbers consist of a real and an imaginary part, which we will use to represent the contribution of cosine and sine waves, respectively, to a given function. For example, 5 + 3i means Re = 5 and Im = 3.

Importantly, the exponential function represents a rotation in the 2D space spanned by the real and imaginary axes:

\[
e^{i\alpha} = \cos \alpha + i \sin \alpha
\]

2D Discrete Fourier Transform

What do these equations mean? Well, if you look at the inverse transform, you see that each \(\hat{A}_k[l]\) is multiplied by an exponential term for each pixel, and in sum they give us the original image. In other words, each exponential function represents one particular wave, and the sum of these waves, weighted by \(\hat{A}\), is our original image A.

In the exponential term, you see an expression such as \((k/m + l/j)\). This means that the variables k and l determine the “speed” with which the term increases when the spatial positions, i and j, respectively, grow.

Since this term determines the current angle of a wave, k and l determine the frequency of the waves in horizontal and vertical direction, respectively.
Fourier Transform: Rectangle

Fourier Transform: Spatial Shifting

Fourier Transform: Rotation

Fourier Transform: Multiplicity

Fourier Transform: Real-World Images