Tutorial 2 - 1D Fourier Transform

Gidon Rosalki

2025-10-29

1 Introduction

A chap called Fourier theorised that any periodic function (signal) can be decomposed into F, a set of sin and cos periodic functions, of different frequencies (called a Fourier series). Since this is not a mathematical course, we will not prove it, nor be expected to understand all the mathematical depths.
Given the set F, we can reconstruct f without any loss of data. The Fourier Transform allows us to use the unique properties of of decomposition into sin and cos on any function.

Given a complex number $c = a + bi = R \cdot e^{ialpha}$, where $e^{i\alpha} = \cos(\alpha) + i\sin(\alpha)$. The absolute value is computed

$$|c| = R = \sqrt{a^2 + b^2}$$

The *phase* is α where

$$\alpha = \tan^{-1}\left(\frac{b}{a}\right)$$

And the *conjugate*:

$$\overline{c} = c^* = a - bi$$

2 Periodic functions

A periodic function can be described sing its **frequency**, **amplitude**, and **phase shift**. The **frequency** is how much time it takes to complete a *complete* phase (so for sin: 2π). The **amplitude** is the absolute maximum value of the wave. The **phase shift** is how much the wave is shifted from its normal position.
The **wavelength** (λ) is the distance over which a periodic wave's shape repeats. As a result, we can see that, since the frequency is the number of waves per unit:

$$f = \frac{1}{\lambda}$$

So, for sin, the wavelength is 2π, and the frequency is $\frac{1}{2\pi}$, or in general for $\sin(ax)$, there is a wavelength of $\frac{2\pi}{a}$, and a frequency of $\frac{a}{2\pi}$. To save effor, we will normally put 2π inside the sin expression as well: $\sin(2\pi\omega x)$, and have a frequency of ωHz, with a wavelength of $\frac{1}{\omega}$.

3 Back to Fourier

By using the Fourier transformation, we are moving from the *time* domain, to the *frequency* domain. We perform a decomposition into different resolutions, the low frequencies give us a rough general structure, and the high frequencies the fine detail. This is very useful for signal understanding, and processing, for jobs like filtering, removing noise, compression, and so on.

3.1 How do we do this?

We want to decompose $f(x)$ into a set of sin and cos functions of different frequencies:

$$f(x) = \sum_{\omega} a_\omega \cos\left(\frac{2\pi\omega x}{N}\right) + b_\omega \sin\left(\frac{2\pi\omega x}{N}\right)$$

We need to find a_ω and b_ω, and we have N data points. This is done with the Fourier transformation.

3.2 Discrete Fourier Transform (DFT)

DFT:

$$F(\omega) = \sum_{x=0}^{N-1} f(x)\, e^{-\dfrac{2\pi i x\omega}{N}}$$

Now is a good time to remember Euler's formula:

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

So:

$$\begin{aligned}
F(\omega) &= \sum_{x=0}^{N-1} f(x)\, e^{-\dfrac{2\pi i x\omega}{N}} \\
&= \sum_{x=0}^{N-1} f(x)\left(\cos\left(-\frac{2\pi x\omega}{N}\right) + i\cdot\sin\left(-\frac{2\pi x\omega}{N}\right)\right) \\
&= \sum_{x=0}^{N-1} f(x)\left(\cos\left(\frac{2\pi x\omega}{N}\right) - i\cdot\sin\left(\frac{2\pi x\omega}{N}\right)\right) \\
&= \sum_{x=0}^{N-1} f(x)\cos\left(\frac{2\pi x\omega}{N}\right) - i\sum_{x=0}^{N-1}\sin\left(\frac{2\pi x\omega}{N}\right)
\end{aligned}$$

And in this beautiful final expression, the first half (the sum over cos) expresses a_ω, and the second half (i times the sum over sin) expresses b_ω.

We also have the IDFT (Inverse DFT):

$$f(\omega) = \frac{1}{N}\sum_{\omega=0}^{N-1} F(\omega)\, e^{\dfrac{2\pi i x\omega}{N}}$$

In real life, we use the Fast Fourier Transform (FFT), which takes only $O(N\log N)$, instead of $O\left(N^2\right)$

3.3 Properties

The FT is linear:

$$\Phi(f(x) + g(x)) = \Phi(f(x)) + \Phi(g(x))$$
$$\Phi(a\cdot f(x)) = a\cdot\Phi(f(x))$$

Additionally, there is scaling: If $f(x)$ gives us the Fourier set $F(u)$, then

$$f(ax) \to \frac{1}{|a|}F\left(\frac{u}{a}\right)$$

Periodicity:

$$\forall k\in\mathbb{Z}\ F(u) = F(u + kN)$$

Symmetry:

$$F(-u) = F^*(u)$$
$$|F(u)| = |F(-u)|$$

So:

$$F(u) = F^*(-u) = F^*(N-u)$$
$$|F(u)| = |F(-u)| = |F(N-u)|$$

What happens if we take $F(0)$? Well:

$$\begin{aligned} F(u) &= \frac{1}{N}\sum_{x=0}^{N-1} f(x)\, e^{\dfrac{-2\pi i u x}{N}} \\ F(0) &= \frac{1}{N}\sum_{x=0}^{N-1} f(x)\, e^{\dfrac{-2\pi i 0 x}{N}} \\ &= \frac{1}{N}\sum_{x=0}^{N-1} f(x) \\ &\approx \text{ The signal average} \end{aligned}$$

Since the FT returns complex numbers:

$$F(u) = R(u) + i \cdot I(u)$$

To visualise, we use the amplitude:

$$|F(u)| = \sqrt{R^2(u) + I^2(u)}$$

3.4 Basis vectors

DFT is a basis transform – We are moving from the standard basis to the Fourier basis. So, this can be done by a matrix multiplication:

$$F(\omega) = \sum_{x=0}^{N-1} f(x)\, e^{-\dfrac{2\pi i x \omega}{N}} \leftrightarrow \vec{F} = M_{N\times N}\vec{f}$$

Where

$$\vec{f} = \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix}$$

and similarly for F. Reminders of this content can be found in algorithms.

4 Non-stationary signals

A stationary signal has the same function, the entire time, but a non stationary signal changes function over time. The Fourier transform provides localisation in the *frequency* domain, but no localisation in time. In short, the resultant graph shows peaks at frequencies, but not when they happen. A non stationary signal will, as a result, produce a noisy Fourier transform, where a stationary one will not.

5 Sound

Sound is a vibration that typically propagates as an audible wave of pressure, through a transmission medium such as a gas, liquid or solid. We represent sound as a stationary, 1D, signal.

Image and sound representation are not very different:

- Quality:
  - Image quality depends on resolution, the number of pixels in the image
  - Sound quality depends on the sampling rate (samples per second)
- Memory:
  - Images are represented using 8 bits per pixel (values between 0, and 255)
  - In sound the bit depth (bits/sample) is defined as the amplitude resolution, so telephone and AM radio are 8 bits, -127 to 127, and audio CDs are 16 bits (-32768, 32767)

6 STFT

When applying DFT on non-stationary signals, localisation in time is lost. To avoid this, we would like to apply DFT independently on short-time segments. We do this by assuming that the segments are short enough to be considered stationary. This is called the Short-Time Fourier Transform (STFT). This is done by the following steps:

1. Choose a window of finite length
2. Place the window on top of signal at $t = 0$
3. Truncate the signal using this window
4. Compute DFT of the truncated signal, save results
5. Incrementally slide the window to the right
6. Go back to step 3, until window reaches the end of the signal

Windows have various properties. Their shape for example, we often use rectangular, Gaussian, triangles, etc. The *length* W, is the size of the window. A longer length results in better frequency resolution, where a shorter length results in better time resolution. Finally, the *shift* L describes how much the window moves every time. A smaller shift results in smoother results, but a larger shift has less computation.

So, the STFT for time t is as follows:

$$F(\omega, t) = \sum_{x=-\infty}^{\infty} f(x) W(t-x) e^{-\frac{2\pi i x \omega}{N}}$$

and the ISTFT (where K is a normalisation constant):

$$f(x) = K \sum_{p=-\infty}^{\infty} \sum_{u=0}^{N-1} F(u, pL) e^{\frac{2\pi i u x}{N}}$$

7 Spectrograms

These are a visual representation of the spectrum of frequencies of a signal as it varies with time. A In a 3D representation, the X axis represents time, Y the frequency, and the non existent Z is instead represented by colours for the amplitude / magnitude. This helps us represent the localisation in time we achieved from STFT.

Sometimes, visualisation is not clear, due to the output range of the transform, which can give us very large values. So, we want to compress the range. The log transform enables us to compress the dynamic range of the values, which has the steps

- Compute $\log(|F(u)| + 1)$
- Scale to the full grey level range