1 Fourier transform and the FFT

Here we write some of the properties of the Fourier transform and the FFT that Matlab uses to calculate the discrete Fourier transform. Matlab uses the Fourier transform convention equivalent to the continuous integral transform (on an infinite domain)

\[
\hat{f}(k) = \int_{-\infty}^{\infty} f(x)e^{-2\pi ikx} \, dx
\]

(1)

\[
f(x) = \int_{-\infty}^{\infty} \hat{f}(k)e^{2\pi ikx} \, dk
\]

(2)

The discrete version of the Fourier transform, the FFT and its inverse, used by Matlab is the same as the standard FFT algorithm in Numerical Recipes. The function uses an FFT algorithm when the input signal is of length \(2^J\), with \(J\) an integer, and uses slower methods when it is not. Matlab’s FFT calculates the equivalent of the discrete sum

\[
F_k = \sum_{n=1}^{N} f_n e^{-i2\pi(k-1)(n-1)/N}, \quad 1 \leq k \leq N
\]

(3)

with \(F_k\) the discrete Fourier transform of a discrete signal \(f_n\) (with index \(n = 1, 2, \ldots, N\)). The inverse transform is given by

\[
f_n = \frac{1}{N} \sum_{k=1}^{N} F_k e^{i2\pi(k-1)(n-1)/N}, \quad 1 \leq n \leq N
\]

(4)

The Fourier transform is defined on the interval \([0, 1)\) and the result is automatically periodized to this interval [due to the \(\exp(i2\pi n)\) periodicity].

The Fourier transform can be computed using the Matlab code

\[
Fk = \text{fft}(fn)/N; \quad \% \text{The FFT of the signal fn}
\]

\[
fn = \text{ifft}(Fk)*N; \quad \% \text{The Inverse FFT}
\]

\[
\text{ifft(fft(fn))}; \quad \% \text{This result = fn (within machine precision)}
\]

We must divide by the number of grid points \(N\) if we want to use the results of the FFT, because Matlab puts the normalization only in the inverse transform. As a result we must then multiply the IFFT by \(N\) to counter Matlab’s dividing by \(N\).

For the 1D FFT, Matlab returns a vector containing the coefficients of the FFT. The result of the FFT is on the k-space interval \([-N/2, N/2]\). However, we must be careful with the way Matlab stores the results of the FFT. The frequencies are stored \([0, 1, 2, \ldots, (N/2 - 1)]\).
Thus the second half of the vector contains the coefficients corresponding to the frequencies in wrap-around order. The frequencies of the computed FFT:

```matlab
% prepare frequencies for plot of Fourier transform (FFT) of the wavelet
k = [1:N/2-1];
k = [0 k -N/2 -fliplr(k)];
delta = 1/N; % L/N, but our interval length is L = 1 here
fk = k/N/delta; % real frequencies

% note that the interval here is [0,1)
```

Because the FFT is discrete, the resulting frequencies are also discrete. The sampling of the signal in physical space will set the limit of the interval of the FFT coefficients (thus the maximum frequency). If sampling of the signal in physical space is $\Delta x$, then the maximum frequency will be $1/(2\Delta x)$. The factor of 2 comes into play due to Nyquist (thus the results are on the interval from $[-N/2, N/2]$). Similarly, the length of the interval of the signal in physical space will set the resolution (sampling interval) of the FFT. If the length of the interval in physical space is $L$, then the resolution of the Fourier transform will be $\Delta k = 1/L$.

Thus both the discreteness of the signal and its finite length are related in the dual spaces. The difference between the discrete transform and the continuous transform should be well noted! The discrete transform inherently maps the result into a finite interval, based upon the resolution of the sampling (even if the function is continuous to begin with). And the transform of signal on a finite interval maps the result onto a discrete interval (even if the finite length function is continuous to begin with).

### 1.1 Effect of translations and dilations

We will often switch between the physical space and Fourier space representation of a signal. Therefore we must keep in mind the effect of translations and dilations of a signal in its dual space.

Dilations of a signal have the following effect

\[
\frac{1}{|a|} f\left(\frac{x}{a}\right) = \hat{f}(ak) \tag{5}
\]

Thus dilations in physical space correspond to contractions in Fourier space, and vice versa.

Translations of a signal in physical space have the following effect

\[
f(x - b) = \hat{f}(k)e^{2\pi i kb} \tag{6}
\]

---

1. Note that the matlab documentation does not explicitly say the order, however this is likely how it is stored based upon the behavior of the `fftsihift` function (swapping of the left and right halves of the vector). In practice mode $N/2$ and mode $-N/2$ are equivalent so that the only difference is in terms of where it is stored.
Translations in physical space corresponds to a phase shift in Fourier space. (from Numerical Recipes)