Geol 335.3

Lab #7: Fourier Transforms and Filtering

Many signals represent functions of a single variable, such as ground motion or sound recorded at a given point is a function of time. In this exercise, you will learn that these records can also be represented in a conjugate domain of variable *frequency* = 1/*time*, as a combination of multiple harmonic functions of time. An example of such combination of cosine functions giving a pulse-like source waveform was used in the preceding labs.

The transformations between the time-domain and frequency-domain representations of a signal are called the forward and inverse Fourier transforms. These transformations are the key part of numerous signal-processing algorithms. In this lab, you will familiarize yourself with Fourier transforms, examine properties of simple signals, and try simple bandpass filtering.

Fourier transforms

Consider any function $u(t)$ of one variable (for example, time t) discretized at N points t_k (for example, $t_k = k\Delta t$, with $k = 0, \dots N-1$). This function can be equivalently represented in two ways:

- In the time domain – as a time sequence of readings $u_k = u(t_k)$. In actual measurements, these readings are real numbers, but generally, u_k can be complex-valued.
- In the frequency domain – as a sum of harmonic functions $\cos(2\pi f_k t)$ and $\sin(2\pi f_k t)$, with some coefficients a_k and b_k, respectively.

The simplest description of transformations between the time- and frequency-domain forms is obtained by using the complex exponent function $\exp(\pm 2\pi i f_k t)$ instead of the cos(…) and sin(…). The frequency-domain coefficients then become complex numbers denoted U_k. These coefficients can be separated into real parts $\mathrm{Re}U_k$ (sometimes called "in-phase" for electric signals) and imaginary parts $\mathrm{Im}U_k$ (called "quadrature"): $U_k = \mathrm{Re}U_k + i\,\mathrm{Im}U_k$.

The frequency-dependent series $\{U_k\}$ (with $k = 0, \dots N-1$) is called the complex spectrum of the signal. From this complex spectrum, the amplitude and phase spectra at frequency f_k are obtained:

- Amplitude spectrum: $Amplitude(f_k) = A_k = |U_k|$;
- Phase spectrum: $Phase(f_k) = \varphi_k = \mathrm{Arg}U_k$.

The phase (argument of a complex number) can be measured in radians (within the $\varphi_k \in (-\pi, \pi]$ or $\varphi_k \in [0, 2\pi)$ range) or in degrees (usually within $\varphi_k \in (-180°, 180°]$ or $\varphi_k \in [0°, 360°)$). The spectral amplitude (magnitude of the complex spectrum) is a nonnegative real number: $A_k \geq 0$.

The spectral amplitude A_k is often strongly variable, which makes it difficult to display its values and compare them at different frequencies. To plot amplitudes, logarithmic (decibel) scaling is used, by plotting the following quantity:

$$\mathrm{dB\_measure}(A_k) = 20\log_{10}\left(\frac{A_k}{A_{\mathrm{ref}}}\right), \tag{1}$$

where A_{ref} is some reference amplitude level. For this reference level, the peak amplitude $A_{\mathrm{ref}} = \max_k (A_k)$ or simple arbitrary selection $A_{\mathrm{ref}} = 1$ are often used.

The time-domain values $\{u_k\}$ above represent the values of the signal at times $t_k = k\Delta t$, where Δt is the sampling interval:

$$\Delta t = \frac{T}{N} = \frac{1}{f_s}, \tag{2}$$

and where f_s is the sampling frequency and $T = N\Delta t$ is the total time duration of the record. Similarly, the spectral quantities $\{U_k\}$, $\{A_k\}$, and $\{\varphi_k\}$ are sampled at N frequencies equally spaced at the frequency sampling interval Δf:

$$\Delta f = \frac{f_s}{N} = \frac{1}{T}. \tag{3}$$

If the signal $\{u_k\}$is real-valued (as always with physical measurements), then half of the values in $\{U_k\}$ with $k > N/2$ are complex conjugates to the other half. Therefore, only $\{U_k\}$ values with $f_k = k\Delta f < f_N$ are mutually independent, where f_N is the Nyquist frequency:

$$f_N = \frac{f_s}{2} = \frac{1}{2T} = \frac{N}{2}\Delta f . \tag{4}$$

This is the largest frequency recoverable by playing back a record discretized at sampling interval Δt.

Because of the significance of the Nyquist frequency f_N as the largest recoverable frequency, it is convenient to sample the frequency domain so that f_k ranges from $–f_N$ to f_N, with zero frequency in the middle: $f_k = k\Delta f – f_N$.

$$f_k = k\Delta f - f_N \, . \tag{5}$$

This is how the frequency sampling is done in this lab. With this sampling, spectral values at negative frequencies are complex conjugates of the values at corresponding positive frequencies: $U\left(-f_k\right) = U^*\left(f_k\right)$.

Equation (2) leads to the very important "time- frequency uncertainty relation":

$$T \cdot \Delta f = 1 \, . \tag{6}$$

This relation means that both time and frequency of the signal cannot be localized simultaneously. If some signal is localized within a small time interval T, then in the frequency domain, it occupies a broad frequency range $\Delta f \approx 1/T$. Vice versa, if we want to record a signal with narrow bandwidth Δf, then we must use a long recording time: $T \approx 1/\Delta f$.

With any time and frequency sampling $\{t_k\}$ and $\{f_k\}$The forward and inverse Fourier transforms allow calculating the $\{U_k\}$ frequency series from the time series$\{u_k\}$ and vice versa:

$$U_k = \sum_{j=0}^{N-1} e^{-i2\pi f_k t_j} u_j \, , \tag{7}$$

$$u_j = \frac{1}{N} \sum_{k=0}^{N-1} e^{i2\pi f_k t_j} U_k \, , \tag{8}$$

where $t_j = j\Delta t$ is the time of the jth sample. These relations show that both $\{u_k\}$and $\{U_k\}$ contain the same information (describe the same seismic signal), but in different forms which allow seeing different aspects of the data.

Fast Fourier Transform (FFT)

In computer codes, the multiplications and summations in eqs. (7) and (8) are implemented in a clever recursive sequence of operations called the Fast Fourier Transform, or FFT. This algorithm evaluates the exact sums (7) or (8) but is much faster than their straightforward summation. For N sampling times (and frequencies), eqs. (7) and (8) contain N sums with N terms in each, that is an order of N^2 operations. However, by utilizing properties of exp(…) functions, the FFT algorithm evaluates all these sums by using ~$N\log N$ operations, which is much fewer than N^2.

FFT tool

Copy to yourself and unpack the zipped archive in lab directory (folder). Open Matlab in that directory and execute script `fftlab.m`. The script should give a large window with six plot panels (produced by `subplot` command in Matlab). If the window is too large or too small for your screen, you can adjust the dimensions by changing values of field '`Position`' in the call of function `figure()` in script `fftlab.m`. The four position values shown like '[100,100,1600,1800]' are the coordinates of the lower-left corner of the window on your screen, and the window width and height, in screen pixels. You can also search Internet for a method to obtain your full screen size automatically (this is very easy).

The two panels on the left of the figure are the real and imaginary parts of the signal in the time domain. The time sampling interval is 1 ms, and the total record is 32 ms long. In practice, these signals would usually contain several thousand of time or frequency points, but we work with only a small example for clarity.

The two panels in the middle of the figure are the real and imaginary parts of the same signal in frequency domain. The panels on the right (with red symbols) show the amplitude and phase spectra. Zero time is shown in the middles of the time panels, and zero frequency is marked by the open circle in the bottom.

For simplicity of displays, all plots are normalized to unit peak values of the complex signal magnitudes. The scaling values are printed in the titles of the upper subplots.

Operate the script by pressing keyboard keys 'c' (to clear the plots) and 'q' (to exit), and the left and right mouse buttons to construct the desired waveforms or spectra. Clicking the left mouse button allows picking intervals of times or frequencies, and the right mouse button gives individual points.

A brief summary of the available commands is printed along the bottom of the window.

Assignments

Perform the following tests and answer questions. Make plots of your work by using figure menu operations in Matlab.

1) (8%) Note the total frequency range. How does it relate to the time sampling interval (1 ms)? What is the sampling interval in the frequency domain?

Note that the frequency axis in the middle panels extends from $–f_N$ to f_N, where f_N is the Nyquist frequency (highest frequency reproducible by discretization). Therefore, the total extent of the frequency domain is the sampling frequency $f_s = 2f_N$, which we have discussed in class and homework assignments.

2) (8%) Using the left mouse button, **put in a 2-ms wide "*boxcar*" function** ("square wave", or a box of 3 samples of equal values) centered on the origin. Print the plot out and mark it up.

Look at the Fourier transform of this signal (shown in the middle plots and spectra on the right). **How wide is the main lobe** if counting from zero frequency? Note that the physical frequencies which we measure are the non-negative ones, and negative frequencies are just results of "frequency folding".

3) (8%) Press 'c' to clear the displays and **create another boxcar function** with double width of the square-wave function. How wide is the frequency-domain transform now? Repeat with a yet wider boxcar. **What property of the Fourier transform** does this test illustrate? *Hint*: refer to the discussion of time-spectral uncertainty after eq. (3) above.
4) (8%) Clear the display and **create another boxcar function** of the same width but shifted to the left, with its rightmost sample at the time origin. How does the transform compare to the one before? Describe how the amplitude and phase spectra of the frequency-domain signal change.
5) (8%) Clear again and use the right mouse button to **put a single impulse** at the origin. What does its transform look like?
6) (8%) **Move the impulse** 2-5 ms away from the time-zero grid point. **Describe** what happens to the amplitude and phase of the frequency-domain signal. **Compare results** to the observations of step 4).
7) (8%) Clear again and **put in two equal pulses equally spaced around the time origin**, at –2 ms and +2 ms times. This is called an even impulse pair. What is characteristic about its transform?
8) (8%) **Make one of the above pulses negative**. This signal is called the odd impulse pair. What does the transform look like?

In the next steps, form signals in the frequency domain (i.e., use harmonic signals) and see how they look in the time domain:

9) (8%) Put a **single pulse** at 0 kHz on the **real part of the frequency domain**. Repeat this test for a pulse at –0.2 kHz or +0.2 kHz. Describe the resulting time-domain signal. Is it real- or imaginary-valued? Is it an even or odd function of time? Is it periodic, with what period? What phases do you see in the bottom-right plot?

10) (8%) Repeat step 9) with **imaginary-valued** pulses in **the frequency domain** and answer the same questions.

11) (8%) Repeat step 9) with **an even impulse pair at** ±0.2 kHz. Answer the same questions.

12) (8%) **Experiment with low-pass, high-pass, or band-pass filtering**. Put one pulse in the time domain, as in steps 5) or 6). By clicking the left mouse button <u>in the upper-right plot</u> (amplitude spectrum), you can construct some shape of the filter. The filter will be shown by a green line. If you need to clear the filter, press key 'f', and then you can pick it again. When the filter is shaped, press 'a' to apply it to both the time- and frequency-domain records.

    Describe in the report how the time-domain and frequency-domain signals, and the amplitude and phase spectra change.

Note that in the frequency domain (middle plots), filtering is applied to both the positive and negative frequencies. This is necessary to make sure that the <u>filtered signal remains real-valued</u> (i.e., physical).

Also note that the filter implemented here is "zero-phase", which means that it does not change the phase spectrum of the signal and only alters its amplitude spectrum. Such filtering does not change (generally) the times of waveform packets on the seismogram but only changes their shapes.

13) (4%) Without changing the filter, try putting two pulses of different amplitudes and press 'a' again. You should see how the filtered pulses overlap in time. If the pulses are well separated, you should see that their filtered shapes are similar to those in step 12). **Is this so**?

    Make more experiments with the transform tool if you like.

Hand in:

Zipped directory or Word or PDF document containing answers to the above questions and images.