Discrete Fourier transform (DFT)

Alejandro Ribeiro

January 23, 2021

Let $x : [0, N-1] \rightarrow \mathbb{C}$ be a discrete signal of duration N and having elements $x(n)$ for $n \in [0, N-1]$. The discrete Fourier transform (DFT) of x is the signal $X : \mathbb{Z} \rightarrow \mathbb{C}$ where the elements $X(k)$ for all $k \in \mathbb{Z}$ are defined as

$$X(k) := \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) \exp(-j2\pi kn/N). \quad (1)$$

The argument k of the signal $X(k)$ is called the frequency of the DFT and the value $X(k)$, the frequency component of the given signal x. When X is the DFT of x we write $X = \mathcal{F}(x)$. The DFT $X = \mathcal{F}(x)$ is also referred to as the spectrum of x. Recall that for a complex exponential, discrete frequency k is equivalent to (real) frequency $f_k = (k/N) f_s$, where N is the total number of samples and f_s the sampling frequency. When interpreting DFTs, it is often easier to consider the real frequency values instead of the corresponding discrete frequencies.

An alternative form of the DFT is to realize that the sum in (1) is defining the inner product between x and the complex exponential e_{kN} with elements $e_{kN}(n) = (1/\sqrt{N}) e^{j2\pi kn/N}$. We can then write

$$X(k) := \langle x, e_{kN} \rangle = \sum_{n=0}^{N-1} x(n) e_{kN}^{*}(n) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) exp(-j2\pi kn/N). \quad (2)$$

This latter expression emphasizes the fact that $X(k)$ is a measure of how much the signal x resembles an oscillation of frequency k.

Because complex exponentials of frequencies k and $k+N$ are equivalent, it follows that DFT values $X(k)$ and $X(k+N)$ are equal, i.e.,

$$\begin{aligned} X(k+N) &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi(k+N)n/N} \\ &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \\ &= X(k) \end{aligned} \tag{3}$$

The relationship in (3) means that the DFT is periodic with period N and while it is defined for all $k \in \mathbb{Z}$, only N values are different. For computational purposes we work with the canonical set of frequencies in the interval $k \in [0, N-1]$. For interpretation purposes we work with the canonical set of frequencies $k \in [-N/2, N/2]$. This latter canonical set contains $N+1$ frequencies instead of N—frequencies $N/2$ and $-N/2$ are equivalent in that $X(N/2) = X(-N/2)$—but it is used to have a set that is symmetric around $k = 0$. Going from one canonical set to the other is straightforward. The frequencies in the interval $[0, N/2]$ are present in both sets and to recover, e.g., the negative frequencies $k \in [-N/2, -1]$ from the positive frequencies $[N/2, N-1]$ we just use the fact that

$$X(-k) = X(N-k), \qquad \text{for all } k \in [1, N/2] \tag{4}$$

We say that the operation in (4) is a "chop and shift." To recover the DFT values for the canonical set $[-N/2, N/2]$ from the canonical set $[0, N-1]$ we chop the frequencies in the interval $[N/2, N-1]$ and shift them to the from of the set. For the purposes of this homework, when you are asked to report a DFT, you should report the DFT for the canonical set $[-N/2, N/2]$.

1 Spectrum of pulses

In this first part of the lab we will consider pulses and waves of different shapes to understand what information can be gleaned from spectral analysis. A prerequisite for that is to have a function to compute DFTs. In the following experiments, you should use the class `cexp` provided on the course website to generate complex exponentials. Type `help (cexp)` in the console to see how this class works.

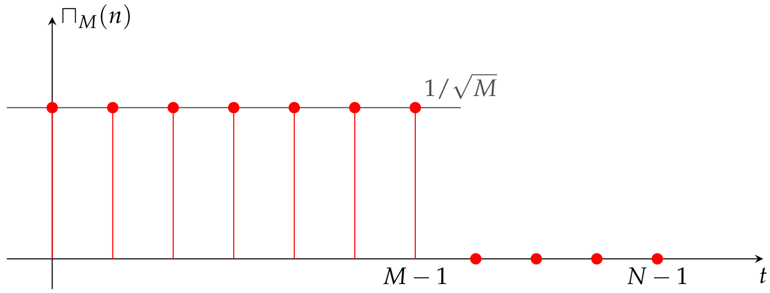

Figure 1. Unit energy square pulse of length $T_0 = MT_s$ and duration $T = NT_s$. The signal is constant for indexes $n < M$ and null for other n. The height of the pulse is set to $1/\sqrt{M}$ to have unit total energy.

1.1 Computation of the DFT. Write a Python class that takes as input a signal x of duration N and associated sample frequency f_s and returns the values of the DFT $X = \mathcal{F}(x)$ for the canonical set $k \in [-N/2, N/2]$ as well as a vector of frequencies with the real frequencies associated with each of the discrete frequencies k. Explain how to use the outcome of this function to recover DFT values $X(k)$ associated with frequencies in the canonical set $k \in [0, N-1]$.

With N samples and a sampling frequency f_s the total signal duration is $T = NT_s$, where $T_s = 1/f_s$ is the sampling period. Given a length $T_0 = MT_s < T$, we define the unit energy square pulse of time length T_0, or, equivalently, discrete length M, as

$$\begin{aligned} \sqcap_M(n) &= \frac{1}{\sqrt{M}} && \text{if } 0 \leq n < M, \\ \sqcap_M(n) &= 0 && \text{if } n \geq M. \end{aligned} \tag{5}$$

Intuitively, pulses of shorter length are faster signals than pulse of longer length. We will see that this rate-of-change information is captured by the DFT.

1.2 DFTs of square pulses. Use the code in Part 1.1 to compute the DFT of square pulses of different lengths with duration $T = 32$s and sample rate $f_s = 8$Hz. You should observe that the DFT is more concentrated for wider pulses. Make this evaluation more quantitative by computing the DFT energy fraction corresponding to frequencies f_k in the interval $[-1/T_0, 1/T_0]$. Report your results for pulses of duration $T_0 = 0.5$s, $T_0 = 1$s, $T_0 = 4$s, and $T_0 = 16$s.

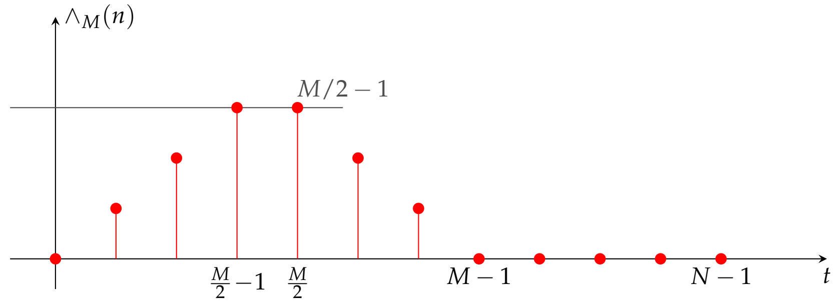

Figure 2. Unit energy triangular pulse of length $T_0 = MT_s$ and duration $T = NT_s$. The signal is smoother, i.e., changes more slowly, than the square pulse of equivalent length.

While it is true that wider pulses change more slowly, all square pulses have, at some point, a high rate of change when they jump from $x(M-1) = 1/\sqrt{M}$ to $x(M) = 0$. We can construct a pulse that changes more slowly by smoothing out the transition. One possibility is to define a triangular pulse by raising and decreasing its height linearly. Specifically, consider an even pulse length M and define the triangular pulse as

$$\begin{aligned} \wedge_M(n) &= n && \text{if } 0 \leq n < M/2, \\ \wedge_M(n) &= (M-1) - n && \text{if } M/2 \leq n < M, \\ \wedge_M(n) &= 0 && \text{if } n \geq M. \end{aligned} \tag{6}$$

Observe that, as defined in (6), the triangular pulse does not have unit energy. In your comparisons below, you may want to scale the pulse numerically to have unit energy. To do so, you just have to divide the pulse by its norm, i.e., use $\wedge_M(n)/\|\wedge_M\|$ instead of $\wedge_M(n)$.

1.3 DFTs of triangular pulses. Consider the same parameters of Part 1.2 and observe that, as in the case of square pulses, the DFT is more concentrated for wider pulses. Make this observation quantitative by looking at the DFT energy fraction corresponding to frequencies f_k in the interval $[-1/T_0, 1/T_0]$. Report your results for pulses of duration $T_0 = 0.5$s, $T_0 = 1$s, $T_0 = 4$s, and $T_0 = 16$s. Compare your results with the results of Part 1.2. Is your observation consistent with the intuitive appreciation that the triangular pulse changes more slowly than the square pulse? A qualitative explanation suffices for most people, but a good engineer would provide a quantitative answer.

1.4 Other pulses. We can define some other pulses with more concentrated spectra. These pulses are also called windows and there is an extensive literature on windows with appealing spectral properties. Find out about Kaiser windows, raised cosine, Gaussian, and Hamming windows. Compare the spectra of one of these windows to the spectra of square and triangular pulses.

2 Properties of the DFT

Our interest in the DFT is, mainly, as a computational tool for signal processing and analysis. For that reason, we will rarely be working on computing analytical expressions. There are, however, some DFT properties that are important to understand analytically. In this part of the assignment we will work on proving three of these properties: conjugate symmetry, energy conservation, and linearity.

2.1 Conjugate symmetry. Consider a real signal x, i.e., a signal with no imaginary part, and let its DFT be $X = \mathcal{F}(x)$. Prove that the DFT X is conjugate symmetric, i.e.,

$$X(-k) = X^*(k) \tag{7}$$

2.2 Energy conservation (Parseval's Theorem). Let $X = \mathcal{F}(x)$ be the DFT of signal x and restrict the DFT X to a set of N consecutive frequencies. Prove that the energies of x and the restricted DFT are the same,

$$\sum_{n=0}^{N-1} |x(n)|^2 = \|x\|^2 = \|X\|^2 = \sum_{k=N_0}^{N_0+N-1} |X(k)|^2. \tag{8}$$

The constant N_0 in (8) is arbitrary.

2.3 Linearity. Prove that the DFT of a linear combination of signals is the linear combination of the respective DFTs of the individual signals:

$$\mathcal{F}(ax + by) = a\mathcal{F}(x) + b\mathcal{F}(y). \tag{9}$$

In (9), both signals are of the same duration N—otherwise, the sum wouldn't be properly defined.

The properties above are very important in the spectral analysis of signals. We present below a fourth property that is not as important, but nevertheless worth knowing.

2.4 Conservation of inner products (Plancherel's Theorem). Let $X = \mathcal{F}(x)$ be the DFT of signal x and $Y = \mathcal{F}(y)$ be the DFT of signal y. Restrict the DFTs X and Y to a set of N consecutive frequencies. Prove that the inner products $\langle x, y \rangle$ between the signals and $\langle X, Y \rangle$ between the restricted DFTs are the same,

$$\sum_{n=0}^{N-1} x(n)y^*(n) = \langle x, y \rangle = \langle X, Y \rangle = \sum_{k=N_0}^{N_0+N-1} X(k)Y^*(k) \tag{10}$$

The constant N_0 in (10) is arbitrary. Observe that the result in Part 2.2 is a particular case of Plancherel's Theorem.

3 The spectra of musical tones

In the first lab assignment we studied how to generate pure musical tones. To do so we simply noted that for sampling time T_s a cosine of frequency f_0 is generated according to the expression

$$x(n) \;=\; \cos\left[2\pi(f_0/f_s)n\right] \;=\; \cos\left[2\pi f_0(nT_s)\right], \tag{11}$$

where the index n varies from 0 to $N-1$, which is equivalent to observing the tone between times 0 and $T = NT_s$. As already noted, the last expression in (11) is intuitive. It's saying that the continuous time cosine $x(t) = \cos(2\pi f_0 t)$ is being sampled every T_s seconds during a time interval of length $T = NT_s$ seconds.

Musical tones have specific frequencies. In particular, the middle A note corresponds to a frequency of 440Hz and the 49th key of a piano. The other 88 basic notes generated by a piano have frequencies that follow the formula

$$f_i = 2^{(i-49)/12} * 440. \tag{12}$$

We have already used this knowledge to play a song using pure musical tones. In this lab assignment, we will compute the DFT of the song we played and interpret the result. For this section only, you should use the class `dft` provided on the course website. Type `help dft` in the console to see how this class works.

3.1 DFT of an A note. Generate an A note of duration $T = 2$ seconds sampled at a frequency $f_s = 44,100$ Hertz. Compute the DFT of this signal and verify that: (a) The DFT is conjugate symmetric and (b) Parseval's

Theorem holds. We know that the DFT of a discrete cosine is given by a couple of delta functions. The DFT of this *A* note, however, is close to that but not exactly. Explain why and find a frequency or frequency range that contains at least 90% of the DFT energy. What can you change to make the spectrum exactly equal to a pair of deltas?

3.2 DFT of a musical piece. Concatenate tones to interpret a musical piece with as many notes as *Happy Birthday.* Compute the DFT of this piece and identify the different musical tones in your piece.

3.3 Energy of different tones of a musical piece. For each of the tones identified in Part 3.2, compute the total energy that the musical piece contains on the tone. Cross check that this energy is, indeed, the energy that you know should be there because of the number of times you played the note.

The rich sound of actual musical instruments comes from the fact that they don't play pure tones, but multiple harmonics. A generic model for a musical instrument is to say that when a note is played it generates not only a tone at the corresponding frequency, but a set of tones at frequencies that are multiples of the base tone. To construct a model say that we are playing a note that corresponds to base frequency f_0. The instrument generates a signal that is given by a sum of multiple harmonics,

$$x(n) = \sum_{h=1}^{H} \alpha_h \cos\Big[2\pi h f_0 (nT_s)\Big]. \tag{13}$$

In (13), H is the total number of harmonics generated by the instrument and α_h is the relative gain of the hth harmonic. The constants α_h are specific to an instrument. E.g., we can get a sound reminiscent of an oboe with $H = 8$ harmonics and gains α_h given by the components of the vector:

$$\alpha = [1.386, 1.370, 0.360, 0.116, 0.106, 0.201, 0.037, 0.019]^T. \tag{14}$$

Likewise, we can get something not totally unlike a flute with $H = 5$ harmonics and gains

$$\alpha = [0.260, 0.118, 0.085, 0.017, 0.014]^T. \tag{15}$$

A very quacky trumpet can be simulated with $H = 13$ harmonics having gains

$$\alpha = [1.167, 1.178, 0.611, 0.591, 0.344, 0.139, \\ 0.090, 0.057, 0.035, 0.029, 0.022, 0.020, 0.014]^T, \quad (16)$$

and an even more quacky clarinet with $H = 19$ harmonics with gains

$$\alpha = [0.061, 0.628, 0.231, 1.161, 0.201, 0.328, 0.154, 0.072, 0.186, 0.133, \\ 0.309, 0.071, 0.098, 0.114, 0.027, 0.057, 0.022, 0.042, 0.023]^T. \quad (17)$$

We can use this harmonic decompositions to play songs with more realistic sounds.

3.4 DFT of an A note of different musical instruments. Compute the DFT of an A note played by each of the 4 musical instruments described above. Determine the frequency range that contains 90% of the energy of the signal.

3.5 DFT of your song on a musical instrument. Compute the DFT of the song selected in Part 3.2 played on *one* of the musical instruments described above. Identify the different musical tones in your piece. If you have no favorite instrument, choose the flute.

4 Time management

The problems in Part 1 are not straightforward but not too difficult. The goal is to finish that up during the first lab session of the week. Try to get a head start in solving the problems. You may not succeed, but thinking about them will streamline the first lab session. This should require 2 more hour outside of the lab session.

The problems in Part 2 will take another couple hours to complete. You should wait until after class on Wednesday to solve them. We will do parts 2.1, 2.2 and 2.3 in class. I am asking you to include them in your report to make sure that you understood them. To solve Part 2.4 you have to work on your own, but the solution is a simple generalization of Part 2.2. You should be able to wrap this up in 2 hours, about 30 minutes for each of the questions.

Part 3 is the one that will take more time because you have to use your problem solving skills. It should take about 6 hours to complete. I would say something like 4 hours for the first three parts and 2 more hours to wrap up the pieces that simulate the wind instruments.

5 Report presentation

Please remember to label both the x- and y-axis, include a legend (if necessary), and add a title to all your figures (check library `matplotlib.pyplot` and commands `xlabel()`, `ylabel()`, `legend()`, and `title()`). A graph without units and labeled axes makes no sense and the titles help us with grading.

Please include your code along with the lab report and don't forget to put the both of your names on the first page. If you are working alone (which you can do only occasionally), state it in the report.