Ph 21.2: Introduction to Fourier Transforms

-v20190418-

Introduction to Fourier transforms

A useful, pragmatic introduction to Fourier transforms (or more generally "spectral methods") is *Numerical Recipes*. This is a very rich subject and *Numerical Recipes* dedicates over 100 pages to the topic of spectral analysis. In this assignment, you will gain some familiarity with the *Fast Fourier Transform* (FFT), one of the most widely used algorithms in all of science and engineering. You will also be introduced to a cousin of the FFT, suitable for analysis of unequally sampled data.

Fourier transforms are one of the most ubiquitous tools of modern physics. One way to describe them is that they allow us to switch back and forth between the description of a physical process as a function of *time* (or position), and the description of the same process as a function of *frequency* (or spatial frequency). Let us however take a step back and introduce the more general definition of *spectral expansion*: the representation of a function $h(x)$ as a sum, with appropriate coefficients, over a set of N *basis functions*:

$$h(x) = \sum_{k=0}^{N-1} \tilde{h}_k \phi_k(x); \tag{1}$$

note that N might be infinite, and that the index k might actually represent a continuous variable (in that case, the symbolic sum $\sum$ should be written as an integral).

Different spectral expansions use different bases, chosen to have certain desired properties: for instance, the Fourier expansion uses functions of definite frequency, the complex exponentials[1] $\phi_k(x) = \exp(-2\pi i f_k x)$, so it is useful to identify the frequency components of a physical process. Consider a *periodic* function $h(x)$ with $h(x+L) = h(x)$; without losing any information, we may then restrict our consideration to the interval $[0, L]$, and write the *Fourier series*

$$h(x) = \sum_{k=-\infty}^{\infty} \tilde{h}_k e^{-2\pi i f_k x}, \quad f_k = k/L, \tag{2}$$

where the *Fourier coefficients* $\tilde{h}_k$ can be recovered as

$$\tilde{h}_k = \frac{1}{L} \int_0^L h(x) e^{2\pi i f_k x}\, dx. \tag{3}$$

For $k = 0$, we get the special case of zero frequency, which represents a finite, constant offset from zero, sometimes known as *DC* offset. For $k = \pm 1$, we get the basis functions $\exp(\pm 2\pi i x/L)$, which have the smallest frequency included in the expansion, $\pm 1/L$; these functions oscillate exactly once in the interval $[0, L]$. The value $1/L$ represents also the difference in frequency between successive basis functions: this is because all basis functions must complete an integer number of periods between 0 and L, and adding $1/L$ to the frequency adds a complete period.

The discrete Fourier transform

Given that this is a computational physics lab, there are two things in Eqs. (2) and (3) that should strike you as inconvenient. First, the function $h(x)$ is reconstructed by means of an infinite series, while we know that in practice we will have to truncate the series at a finite k. Second, the coefficients $\tilde{h}_k$ are expressed formally as analytic integrals over a continuous range of x values, while we know that in practice we will have to use a numerical approximation to the integral, not least because both experimental data and their computer representation will provide only a *finite set* of values $h(x)$, often at *equispaced* abscissae:

$$h_j = h(x_j), \quad \text{with} \quad x_j = jL/N, \quad \text{and} \quad j = 0, \ldots, N-1 \tag{4}$$

[1] Complex-number refresher: $(a+ib)+(c+id) = (a+c)+i(b+d)$, $(a+ib)(c+id) = (ac-bd)+i(bc+ad)$, $1/(a+ib) = (a-ib)/(a^2+b^2)$, $\exp iz = \cos z + i \sin z$, and hence $a + ib = \sqrt{a^2+b^2}\exp(i \arctan b/a)$.

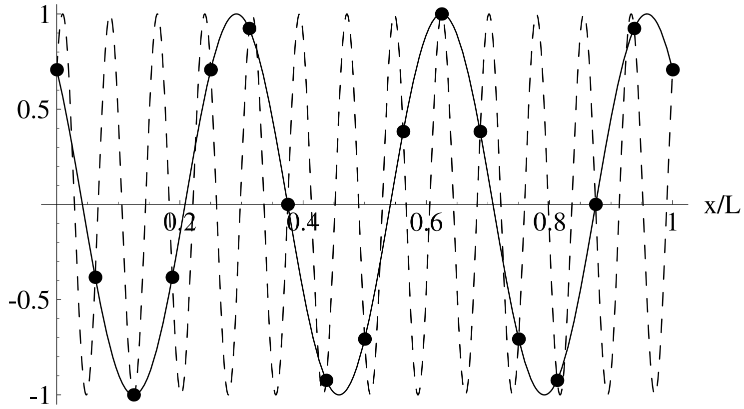

Figure 1: *Function aliasing*: when sampled at $N = 16$, the function $\sin(2\pi \cdot 13x/L + \pi/4)$, with $f = 13/L$, appears as the function $\cos(2\pi \cdot 3x/L + \pi/4)$, with $f = 3/L$. More generally, the frequency $f = P/L$, with $P > N/2$, is *aliased* to $f = (N - P)/L$.

[because the function is periodic, we do not need to define $h_N \equiv h(L)$, which is the same as $h_0 \equiv h(0)$]. We say that the function $h(x)$ is *sampled* with the *sampling rate* N/L.

Now we reason that the highest frequency that we can hope to observe in this dataset is actually $f_c = (N/2)/L$ (the *Nyquist frequency*), which corresponds to functions that oscillate once every two data points; any frequency higher than that will actually appear disguised as a lower frequency (see Fig. 1). The Nyquist frequency identifies the highest-frequency component that can be represented faithfully in the data set; the faster the sampling rate, the higher the Nyquist frequency.

Thus, we can truncate the Fourier series at the Nyquist frequency, setting $M = N/2$; using the simplest possible approximation to the integral (3), we get the transformations[2]

$$h_j = \sum_{k=-N/2}^{N/2-1} \tilde{h}_k e^{-2\pi i f_k x_j}, \quad \tilde{h}_k = \frac{1}{N} \sum_{i=0}^{N-1} h_j e^{2\pi i f_k x_j}, \quad \text{with} \quad x_j = jL/N,\ f_k = k/L. \tag{5}$$

These equations define the *discrete Fourier transform* and its inverse, which map a set of N *physical-space* values h_i into a set of N *Fourier-space* coefficients $\tilde{h}_k$, and backwards. To make these expressions more symmetric, we note that, for the purpose of the inverse transformation ($\tilde{h}_k \to h_i$), the frequency f_k can be replaced by $f_{(k+N)}$ with no change in the h_i; we then write

$$\begin{aligned} h_j &= \sum_{k=-N/2}^{-1} \tilde{h}_k e^{-2\pi i f_k x_j} + \sum_{k=0}^{N/2-1} \tilde{h}_k e^{-2\pi i f_k x_j} \\ &= \sum_{k=-N/2}^{-1} \tilde{h}_{(k+N)} e^{-2\pi i f_{(k+N)} x_j} + \sum_{k=0}^{N/2-1} \tilde{h}_k e^{-2\pi i f_k x_j} \\ &= \sum_{k=0}^{N-1} \tilde{h}_k e^{-2\pi i f_k x_j}. \end{aligned} \tag{6}$$

With this trick, both the h_j and the $\tilde{h}_k$ are defined for indices ranging from 0 to $N - 1$, but we have to keep in mind the following interpretation: the $\tilde{h}_k$ correspond to the zero frequency for $k = 0$, to positive

[2]Note that the sum over k need only run from $-N/2$ to $N/2 - 1$, since the values of $f_{N/2}$ and $f_{-N/2}$ at the points x_j are the same.

frequencies for $1 \leq k \leq N/2-1$, and to negative frequencies for $N/2+1 \leq k \leq N-1$, while the coefficient $\tilde{h}_{N/2}$ corresponds to both $f_c = N/(2L)$ and $-f_c$.

The fast Fourier transform

In computational applications, it is often not so important whether we know how to do something, but whether we know how to do it *fast*. As a matter of fact, the discrete Fourier transform owes its popularity and success to the fact that computer scientists have devised *fast Fourier transform* (FFT) algorithms that can evaluate both transforms,

$$\tilde{h}_k = \frac{1}{N}\sum_{j=0}^{N-1} h_j e^{2\pi ijk/N} \text{ (transform)}, \quad h_j = \sum_{k=0}^{N-1} \tilde{h}_k e^{-2\pi ijk/N} \text{ (inverse transform)}, \tag{7}$$

in $O(N \log N)$ operations instead of the $O(N^2)$ operations that are (apparently) implicit in Eq. (7). This is a dramatic speedup!

Some of the common applications that have been found for FFTs include the data convolution and deconvolution (which are obtained by multiplying Fourier transforms), correlation and autocorrelation (used to turn time-domain data into frequency-domain data for radio interferometry and spectroscopy), optimal filtering (i.e., finding the best way to remove the noise from a signal), power spectrum estimation (i.e., figuring out the frequency of a signal from its time series), digital filtering (as in oversampling CD players), the computation of Fourier integrals (of the form $I = \int_a^b e^{i\omega t} h(t) dt$), and to some extent the *wavelet transform* (often used for image compression). As always, see *Numerical Recipes* for useful pointers.

While this assignment is not the place to go into the goriest details of FFTs algorithms, we can give an outline of the *Danielson-Lanczos lemma*, which is the basis for many of them. We take the set of N values $\{f_j\}$, and divide it into two sets of $N/2$, with respectively even and odd indices; we then rewrite the inverse transform in (7) as

$$\begin{aligned} \tilde{h}_k &= \frac{2}{2N}\sum_{j=0}^{N/2-1} h_{2j} e^{2\pi ijk/(N/2)} + \frac{2e^{2\pi ik/N}}{2N}\sum_{j=0}^{N/2-1} h_{2j+1} e^{2\pi ijk/(N/2)} \\ &= 2\,\tilde{h}_k^{\text{even}} + 2\,e^{2\pi ik/N}\tilde{h}_k^{\text{odd}}, \end{aligned} \tag{8}$$

where the sets of Fourier coefficients $\{\tilde{h}_k^{\text{even}}\}$ and $\{\tilde{h}_k^{\text{odd}}\}$ represent the result of applying the Fourier transform separately to $\{f_j\}^{\text{even}}$ and $\{f_j\}^{\text{odd}}$. This procedure can be repeated, subdividing the $\{f_j\}$ into smaller and smaller sets, until we obtain N separate sets of just one number. Now, the Fourier transform of a single number is trivial (it is just the number itself), so we begin to go backward, using Eq. (8) to assemble larger and larger sets of Fourier coefficients, until we get back to the full $\{\tilde{h}_k\}$. For N data points, the total number of subdivisions is $N \log N$, which then becomes proportional to the computational cost of the FFT. For the details of implementing such a scheme, see *Numerical Recipes*.

The Assignment (Part I)

1. Prove for yourself that the definition of the Fourier series is consistent: show that inserting Eq. (3) in Eq. (2) yields an identity. *Hint:* You may wish to use that
$$\sum_{n=-\infty}^{\infty} \delta(x/L - n) = \sum_{k=-\infty}^{\infty} e^{2\pi kx/L}\ . \tag{9}$$
2. Show that a suitable linear combination of $\exp(-2\pi ix/L)$ and $\exp(2\pi ix/L)$ can represent any function of the form $A\sin(2\pi x/L + \varphi)$, for any amplitude A and phase φ (*hint: you will need to use complex linear-combination coefficients*).
3. Show that for *real* functions $h(x)$, the Fourier coefficients $\tilde{h}_k$ must satisfy the relation $\tilde{h}_{-k} = \tilde{h}_k^*$ (where "*" denotes complex conjugation), so the set of negative-frequency coefficients carries the same information as the set of positive-frequency coefficients.

4. Convince yourself of the *convolution theorem*, which (in one of its forms) says that the Fourier coefficients of the product $H(x) = h^{(1)}(x)h^{(2)}(x)$ are equal to the *convolution product*

$$\tilde{H}_k = \sum_{k'=-\infty}^{+\infty} \tilde{h}^{(1)}_{k-k'}\tilde{h}^{(2)}_{k'}. \tag{10}$$

   What is the graphical interpretation of this product? Sketch a graph of $\tilde{H}_k$ for a smooth $\tilde{h}^{(1)}_k$ centered around $k = 0$, and for a $\tilde{h}^{(2)}_k$ consisting of a single unit pulse at $k = 10$, that is, $\tilde{h}^{(2)}_k = \delta_{k,10}$ (*hint: to prove the theorem, write $h^{(1)}(x)$ and $h^{(2)}(x)$ as Fourier series, expand their product, and then read off the coefficients $\tilde{H}_k$. Try it first with a few nonzero $\tilde{h}^{(1)}$ and $\tilde{h}^{(2)}$.*)

5. Test the numpy or scipy fft functions by computing the Fourier Transform of the cosine function $C + A\cos(ft + \varphi)$ (with f equal to one of the f_k) and of the Gaussian function $A\exp[-B(t - L/2)^2]$ (for B large enough, this function can be considered periodic with $h(0) = h(L) = 0$). First do the transforms analytically, then verify that your code returns the expected results, that the normalization is correct, and that applying the transform and the inverse transform in sequence returns the original function (*hint: you can look up the transform of the Gaussian function, but be wary of the possibility of different normalizations. Otherwise, ask for the help of your TA.*)

The Assignment (Part II)

The `SETI@home` project (`http://setiathome.ssl.berkeley.edu`) uses the computational power of thousands of volunteer-provided, internet-connected workstations to analyze the signals collected at the radiotelescope in Arecibo, Puerto Rico, for messages from extraterrestrial civilizations. The FFT analysis software runs as a screensaver, taking up unused CPU cycles. The survey covers an interval of frequencies centered at 1420 MHz, which corresponds to the 21 cm Hydrogen line that many researchers have suggested as a likely choice for interstellar communication.

We are providing you with two fictitious time series of 32,768 elements, sampled at intervals of 1 ms, that represent the output of the Arecibo radiotelescope for ~ 32 seconds as it tracks over the sky (use the files `/usr/local/src/arecibo1.txt` and `/usr/local/src/arecibo2.txt`). The units are arbitrary, and the frequencies have already been shifted down so that the continuous component corresponds to 1420 MHz, the ± 100 Hz component to 1420 ± 10^{-4} MHz, and so on.

1. First, let's look at the dataset `/usr/local/src/arecibo1.txt`. Plotting it as a function of time reveals no obvious features (try it!): if there is an extraterrestrial signal here, it is buried in noise. However, if the signal has a (roughly) definite frequency, using the FFT will reassemble all its power into a few nearby Fourier coefficients. See if you can find the signal, and obtain its frequency in Hertz (*hint: be sure to use all the data provided, and define the frequency of the signal as the frequency where the magnitude of the FFT is maximum.*)

2. In the previous task, you should have found that the Fourier transform of the first dataset shows a rounded (rather than sharp) peak near the putative signal frequency. We can explain this behavior as follows (if very roughly). The radiotelescope is sensitive only to signals emitted from a minute region in the sky (toward which it is pointing). To scan extended regions of the sky, the telescope is manouvered so that the region of good sensitivity moves slowly across the sky. The effect on the signal received from a source at a fixed position is that the amplitude of the signal ramps up as the source enters the region of good sensitivity, then winds down as the source exits.

   We can model the variation in the amplitude as a Gaussian envelope $\exp[-(t-t_0)^2/\Delta t^2]$ that multiplies the original signal. From task 4 in part I of the assignment, we know that the Fourier transform of the product of two functions is equal to the *convolution* of the two transforms. See if you can interpret the FFT of the first dataset as the Fourier transform of a perfect sinusoid multiplied by a Gaussian envelope. What is, approximately, the time constant Δt of the envelope? (*hint: superimpose plots of the Fourier-transformed envelope, for different Δts, on your FFT. Because we are convolving a*

Gaussian with a delta function, you will need to shift the frequency axis of the envelope to the central frequency of the extraterrestrial signal.)

3. (Optional) Now let's look at the dataset `/usr/local/src/arecibo2.txt`. It is harder to detect the extraterrestrial source contained in this set, because we have introduced a *frequency drift*, which is present when the remote source is accelerating with respect to the Earth. This means that for all purposes the frequency of the source is changing with time (to first approximation, $f = f_0 + D(t - t_0)$, with the effect of spreading the power of the signal over many frequency components. To recover the signal, apply a set of *dechirping filters* by multiplying the time series by $\exp[2\pi i\, D'\, t^2]$, where D' ranges from 0 to 15 Hz/s. Explain how the dechirping filter works, and give an estimate for the frequency interval swept by the original signal (*hint: by trial and error, or by a more systematic search, find the value D' that gives the best* contrast *between signal and noise. To get the true central frequency of the original signal, you need an estimate of t_0, but obtaining that is outside the scope of this assignment. Assume $t_0 = 16.38\,\mathrm{s}$.*). It might be of interest that one of the frequent instructors of the Ph20 sequence developed a generalization of the FFT called the Fast Chirp Transform (FCT). See Jenet & Prince, Phys. Rev. D, 62, 120001(2000).

The Assignment (Part III)

The FFT is an essential tool for both numerical analysis and data analysis. It is often found "under-the-hood" of many other algorithms. An example is the following.

Unequally Sampled Data

The FFT relies on values that are equally sampled in time or some other variable. But for real scientific data this is often not the case: the times at which data are taken may depend on many factors and may not allow equal sampling. An example is a set of astronomical observations of a source over many nights: first, data can only be taken at night, and second, there can be clouds or other circumstances that prevent data taking. What then?

Numerical Recipes discusses the case of unequal data and describes the "Lomb-Scargle" algorithm. While no longer the best or most sophisticated of the algorithms for treating unequally sampled data, it is perhaps the most intuitive, based on minimizing chi-square summed over the samples. For extensions of Lomb-Scargle see e.g. Bretthorst. A useful python package developed by A. Schwarzenberg-Czerny using the *Analysis of Variance (AoV)* algorithm can be found at :

`http://users.camk.edu.pl/alex/#software`

The assignment is the following:

1. Do a search and find a python implementation of the *Numerical Recipes* Lomb-Scargle algorithm. numpy and scipy versions are available.

2. Use the python Lomb-Scargle routine to analyze the Gaussian distribution from Part I of the assignment, and the first arecibo data set, from Part II of the assignment. These are, of course, equally sampled data and are useful for comparison of Lomb-Scargle to the FFT algorith.

3. Now use Lomb-Scargle on an unequally sampled data set. Use the module developed for the Ph22.1 assignment to grab data from the Catalina Real Time Survey, in particular for the source Her X-1, a binary system containing a neutron star. Use Lomb-Scargle to find significant frequencies in the data. The orbital period is 1.70 days. That should be straightforward to detect. However, there are other significant periods as well. Often spurious periods arise as beats between two significant frequencies. Can you think of other significant frequencies that might arise in astronomical data from an observatory?