Problem Set 6

Andrew Lewis

November 12, 2022

1 Shifting an Array With a Convolution

In order to do this, I had an array of values and an integer of how much to shift it. I then placed a 1 in an otherwise empty array in the spot where I wanted to shift it. The x values range from 0 to 100. By convolving these two arrays we got a copy of the first array multiplied with the kronecker delta function, giving us a shifted array. This is shown in the following figure:

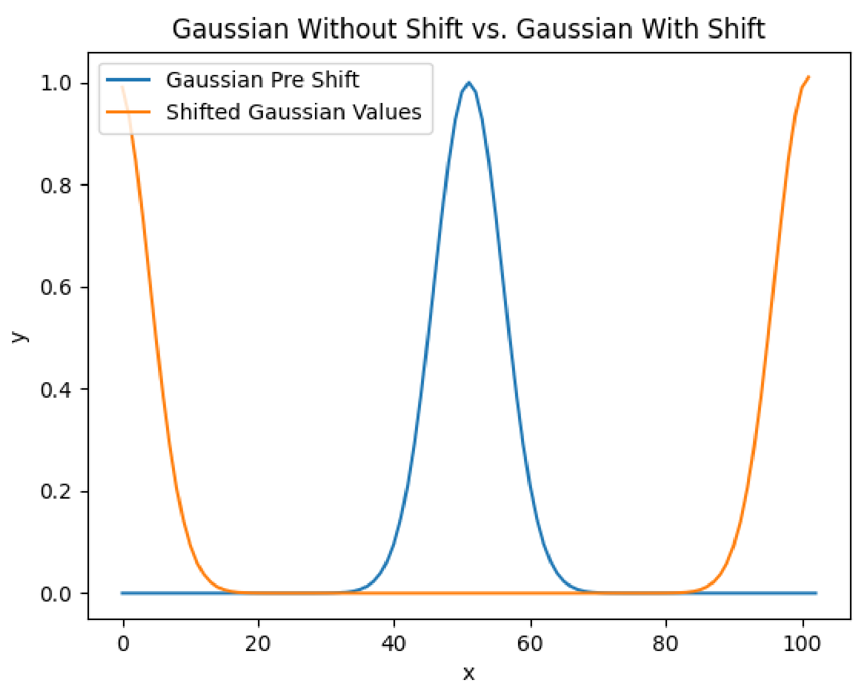

Figure 1: Comparison of a gaussian pre shift vs. post shift. The Gaussian was shifted by half of the array length. The parameters entered into the gaussian were an amplitude of 1, a center at 50, and a variance of 5

2 The Correlation Function

a) Correlation of a non shifted correlation function

The correlation function was computed using ffts with the real and imaginary parts. The x values range from 0 to 100. The plot of a Gaussian correlated with itself is as follows:

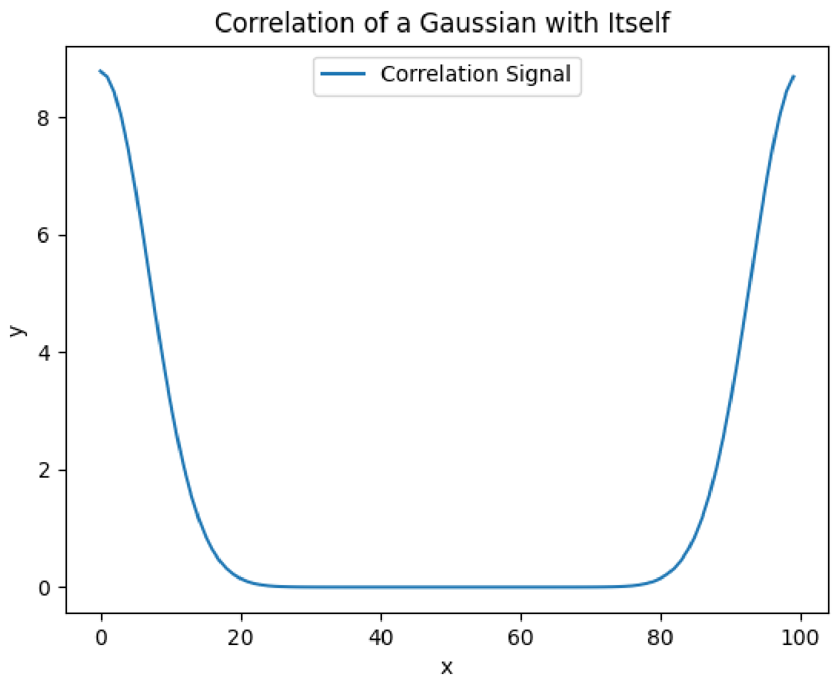

Figure 2: Correlation function of a Gaussian with itself. The parameters passed into the Gaussian were an amplitude of 1, a center at 50 and a variance of 5.

b) Correlation of a Shifted Correlation Function

Taking the correlation of a gaussian with a shifted gaussian results in the following:

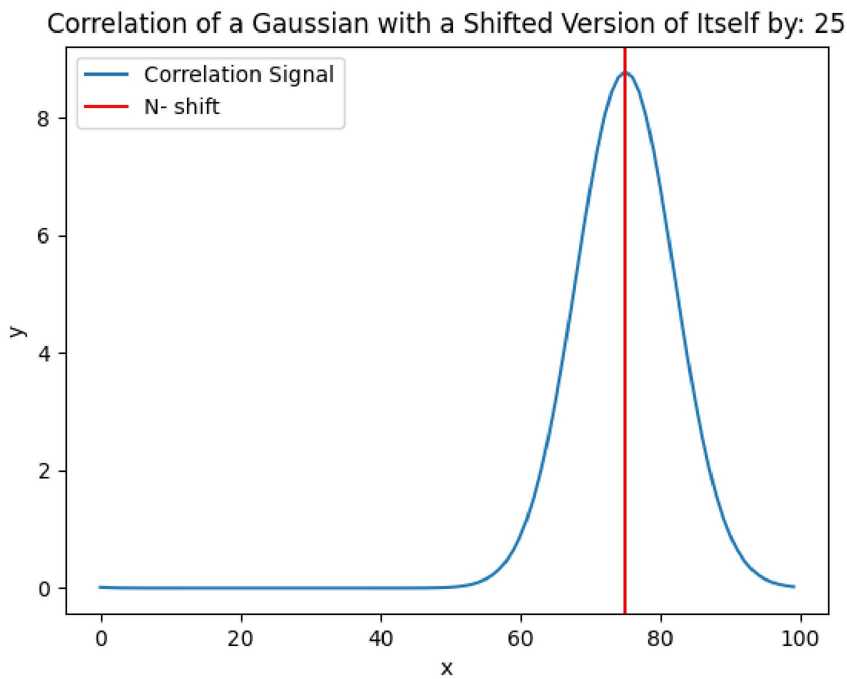

Figure 3: Gaussian correlated with itself shifted by a value of 25. Same arguments passed to the gaussian as question 2a.

As can be seen the correlation of the gaussian with a shifted version of itself gives a gaussian correlation not at 25, but at N - 25. This makes sense that in order for a gaussian to be correlated at 0 again, it would either have to move 25 units backwards or forward 75/100 of a period.

3 Circuluant Nature of the DFT

The code for this section can be found in Q3.py.
Without adjusting for the circulant nature of the DFT the convolution of a 6 unit long ramp with a delta function at the 4th index (x=3) is the following:

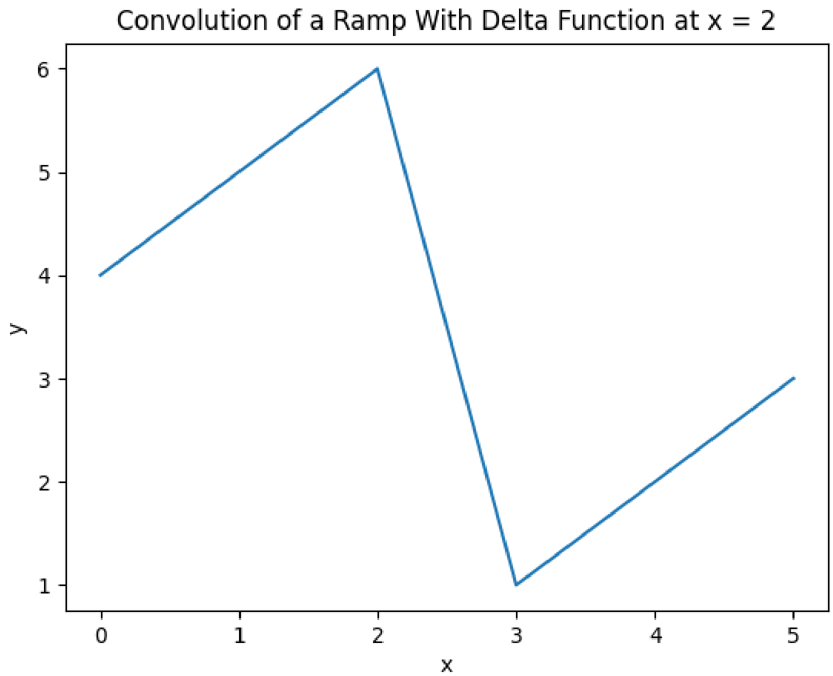

Figure 4: Convolution without adjusting for the wrap around nature of the DFT. As can be seen the ramp bleeds back into the start of the array

After adding zeros to the end of the arrays with the same length as the original arrays, the following convolution can be seen:

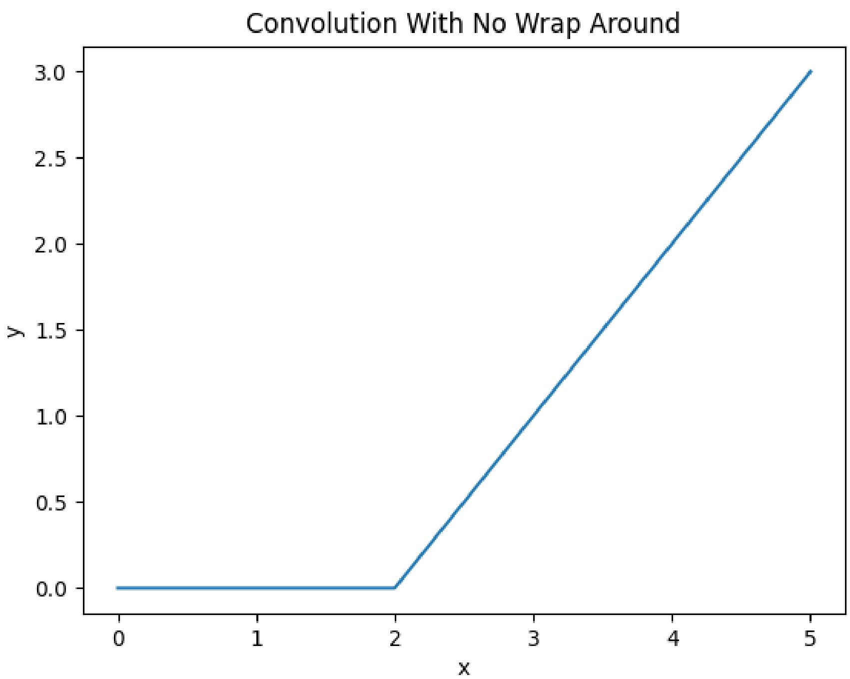

Figure 5: Convolution after adjusting for the circulant nature of the DFT. No more wrap around!

4 Doing Some DFT Math

a)

$$\sum_{x=0}^{N-1} exp(-2\pi ikx/N)$$

See that we can rewrite it as a geometric series, the sum of a geometric series starting at 0 can be written as

$$\sum_{x=0}^{N-1} ar^x = a\frac{(1-r^N)}{(1-r)}$$

So rewriting our sum

$$\sum_{x=0}^{N-1} exp(-2\pi ik/N)^x$$

where

$$r = exp(-2\pi ik/N), a = 1$$

Therefore, plugging it into our formula for the sum of a geometric series

$$\sum_{x=0}^{N-1} exp(-2\pi ikx/N) -> \frac{1-exp(-2\pi ik)}{1-exp(-2\pi ik/N)}$$

b)

Show that this approaches N as k approaches 0
Taking our closed form result from part a), we will manipulate it into a more favorable form:

$$\frac{1 - exp(-2\pi ikx)}{1 - exp(-2\pi ikx/N)} =$$
$$= \frac{e^{-\pi ik}(e^{\pi ik} - e^{-\pi ik})}{e^{-\frac{\pi ik}{N}}(e^{\frac{\pi ik}{N}} - e^{\frac{-\pi ik}{N}})}$$

Which using the following identity

$$2i\sin(x) = e^{ix} - e^{-ix}$$

$$= e^{(-\pi ik)(1-N)}\frac{\sin(\pi k)}{\sin(\frac{\pi k}{N})}$$

Taking the limit of this expression as k $->$ does not yield a satisfactory answer, therefore we need to use l'hopital's rule:

$$\frac{d}{dk}e^{(-\pi ik)(1-N)}\frac{\sin(\pi k)}{\sin(\frac{\pi k}{N})} =$$
$$= -\pi i(1-N)e^{(-\pi ik)(1-N)}\frac{\sin(\pi k)}{\sin(\frac{\pi k}{N})} + e^{(-\pi ik)(1-N)}\frac{\cos(\pi k)}{\cos(\frac{\pi k}{N})}\frac{\pi}{\pi/N}$$

as k goes to zero, the first term will be zero, therefore for the term on the right hand side, plugging in k =0:

$$= e^{(-\pi ik)(1-N)}\frac{\cos(\pi k)}{\cos(\frac{\pi k}{N})}\frac{\pi}{\pi/N}$$
$$= (1)\frac{(1)}{(1)} * N$$
$$= N$$

Therefore as k $->0$ the sum will go to

$$e^{(-\pi ik)(1-N)}\frac{\sin(\pi k)}{\sin(\frac{\pi k}{N})}@(k=0) = N$$

Show the sum is zero for any integer k that is not a multiple of N
We will start with our normal expression

$$\sum = \frac{1 - exp(-\pi ik)}{1 - exp(-2\pi ik/N)}$$

Using the following substitution

$$e^{i}x = cos(x) - isin(x)$$

Our expression becomes:

$$\frac{1 - (\cos(-\pi k) - isin(-\pi k))}{1 - (\cos(-2\pi k/N) - i\sin(-2\pi k/N))}$$

Setting k = N+1 our expression becomes

$$\frac{1-(\cos(-\pi(N+1))-isin(-\pi(N+1)))}{1-(\cos(-2\pi(N+1)/N)-i\sin(-2\pi(N+1))/N))}$$

By inspection, an integer value of $\sin(N\pi)$ will be equal to 0 and an integer value of $\cos(N\pi)$ will be 1, therefore our expression will become:

$$\frac{1-(1)-0}{1-(\cos(-2\pi(N+1)/N)-i\sin(-2\pi(N+1))/N))}$$

And since in the denominator we have non integer values for the arguments of cos and sin we will get:

$$\frac{0}{\text{something non zero}}=0$$

Additionally, for values of N that make the values of cos and sin an integer, recall that

$$\cos(-x)=-\cos(x)$$

Therefore our value will be positive, and since sin is bound between -1 and 1 our denominator will always be non zero

---

It will be zero for any integer k that is not a multiple of N

---

c)

Writing down the DFT of a non-integer sine wave
Let's consider the following sine wave of the form:

$$y(x)=\sin(2\pi x*m/N)$$

the reason we have added the terms of $2*\pi$, $1/N$ are to make our analytic fourier transform work out easier, using euler's formula we can write sin as:

$$\sin(2\pi x*m/N)=(e^{(2\pi ixm/N)}-e^{-(2\pi ix*m/N)})\frac{1}{2i}$$

Now taking the DFT

$$\frac{1}{2i}\sum_{x=0}^{N-1}e^{2\pi ix(m/N)}e^{2\pi i(-k)x/N}-e^{-2\pi ix(m/N)}*e^{-2\pi ikx/N}$$

Simplifying a little

$$\frac{1}{2i}\sum_{x=0}^{N-1}e^{2\pi ix((m-k)/N)}-e^{-2\pi ix((m+k)/N)}$$

Then, using our identity from part a we can write our fourier series as:

$$\frac{1}{2i}\left(\frac{1-e^{2\pi ix((m-k)}}{e^{2\pi ix((m-k)/N)}}-\frac{1-e^{2\pi ix((m+k)}}{e^{2\pi ix((m+k)/N})}\right)$$

We can then plot the DFT and the FFT against each other. For m = 3.5, and N = 100

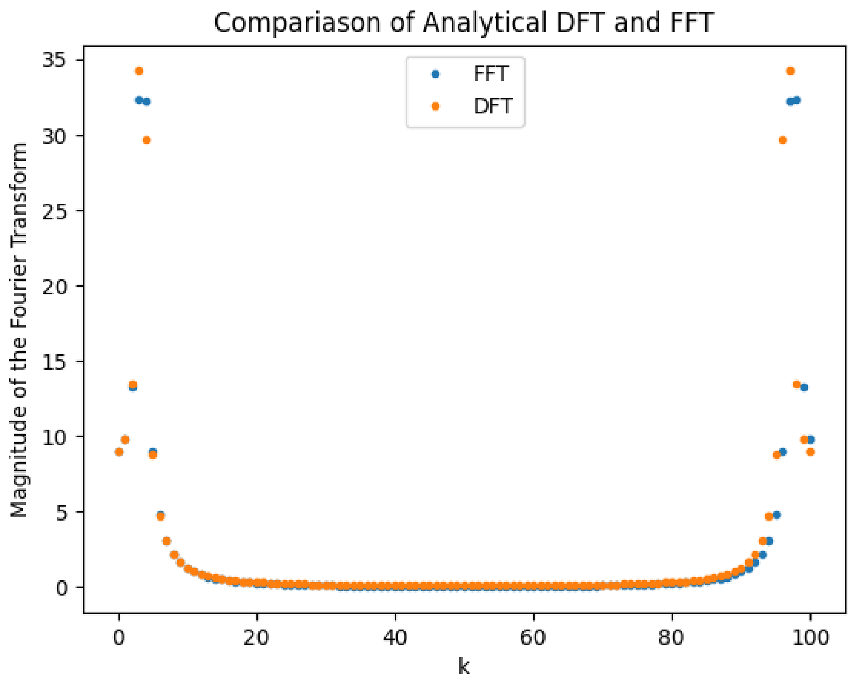

Figure 6: Compariason between the analytical DFT solution of my sine wave and the fourier transform

As can be seen, the two fourier transforms do agree with each other for their general characteristics. However, as can be plainly seen they do not agree with each other to machine precision. Maybe a math error? But I am happy to report that we are close to a delta function!

d)

I used the window function that was suggested in the assignment. Multiplying by this window function yielded the following result:

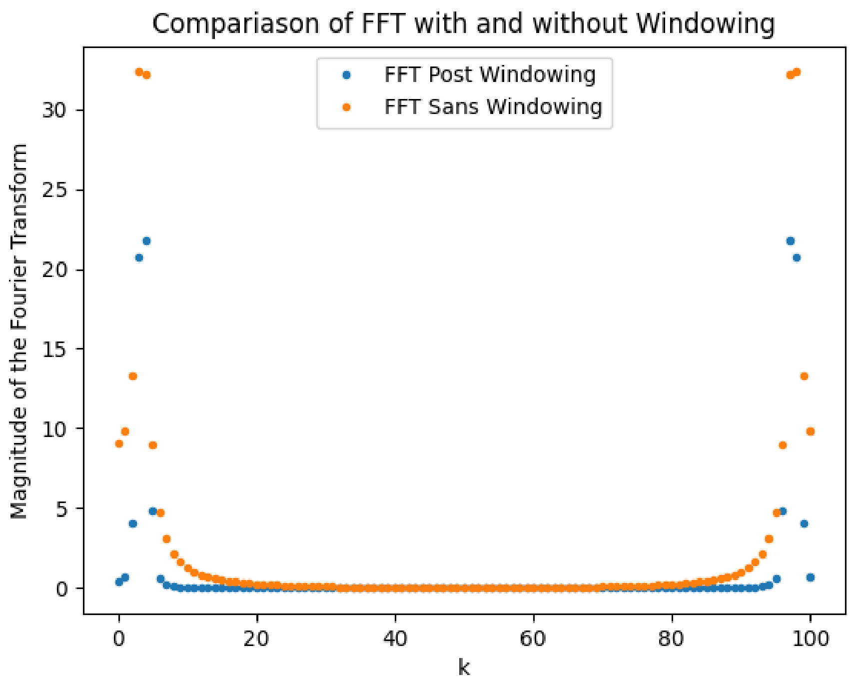

Figure 7: Comparison between the FFT with (Blue) and without (Orange) windowing

As can be seen in the above figure, the FFT with filtering exhibits less spectral leakage, and is therefore a better fourier transform of our sine wave.

e)

The fourier transform of the window, taken with np.fft.fft is:

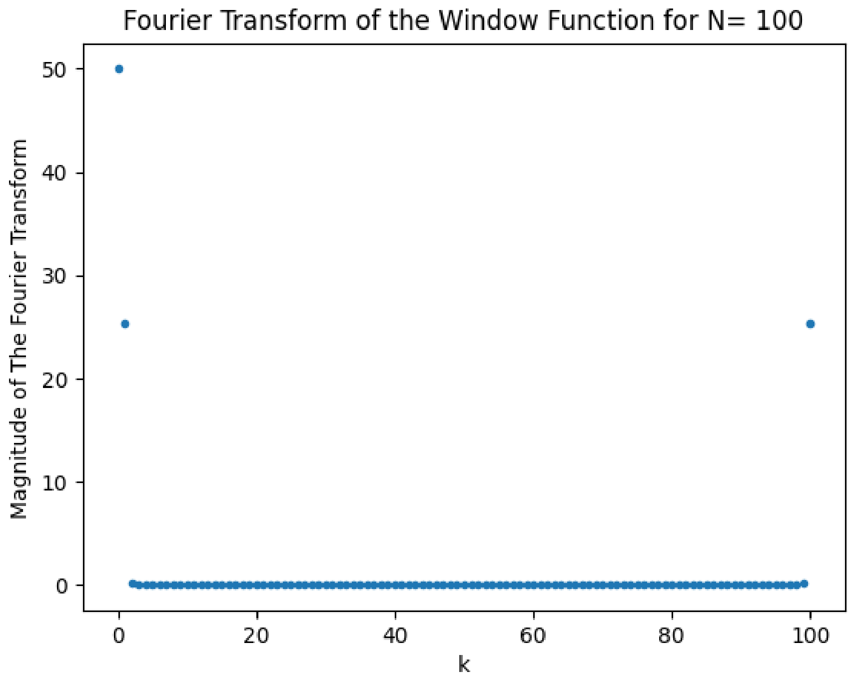

Figure 8: Fourier transform of our window function

As can be seen, for k = 0, the magnitude is 50, for k =1 the magnitude is 25 and for k = 100 the magnitude is also 25, therefore we have shown that the fourier transform of the window has the desired properties.

5 Matched Filter of LIGO data