Problem 1

Choosing to convolve the function with a delta function gives us the desired result. The resulting shift can be seen below.

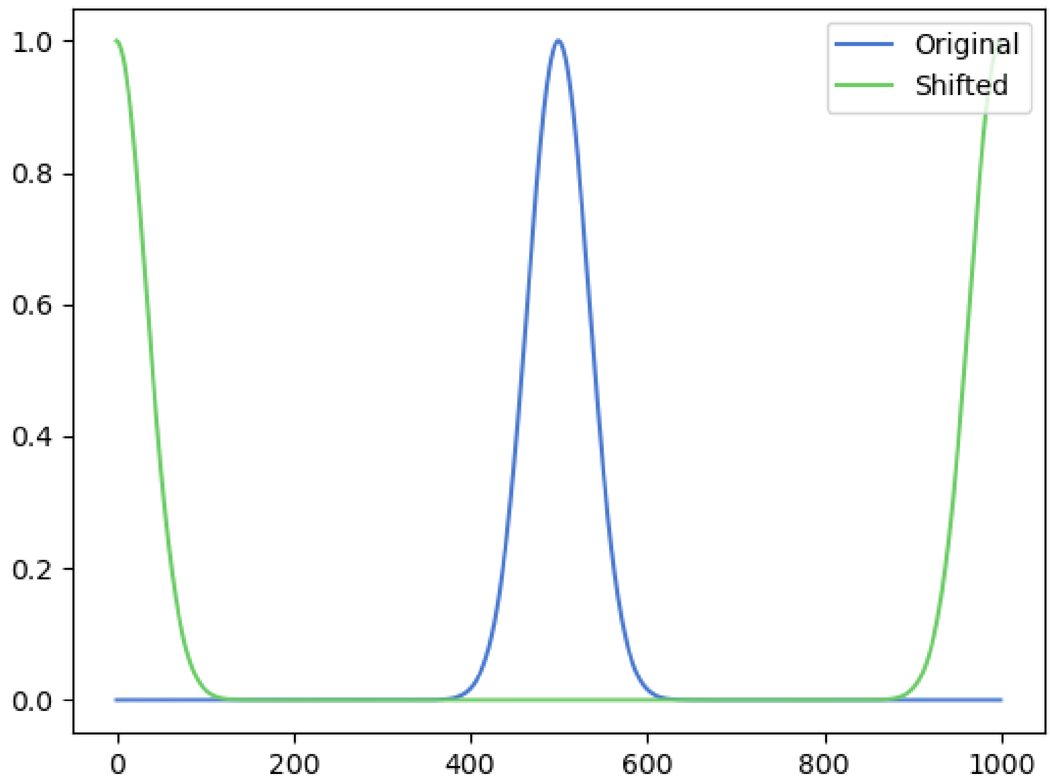

Figure 1: Shifted Gaussian after convolving original function with delta function centered at half length of the input array. Note that this gives us periodic boundaries and the original function wraps around as it approaches the end points.

Problem 2

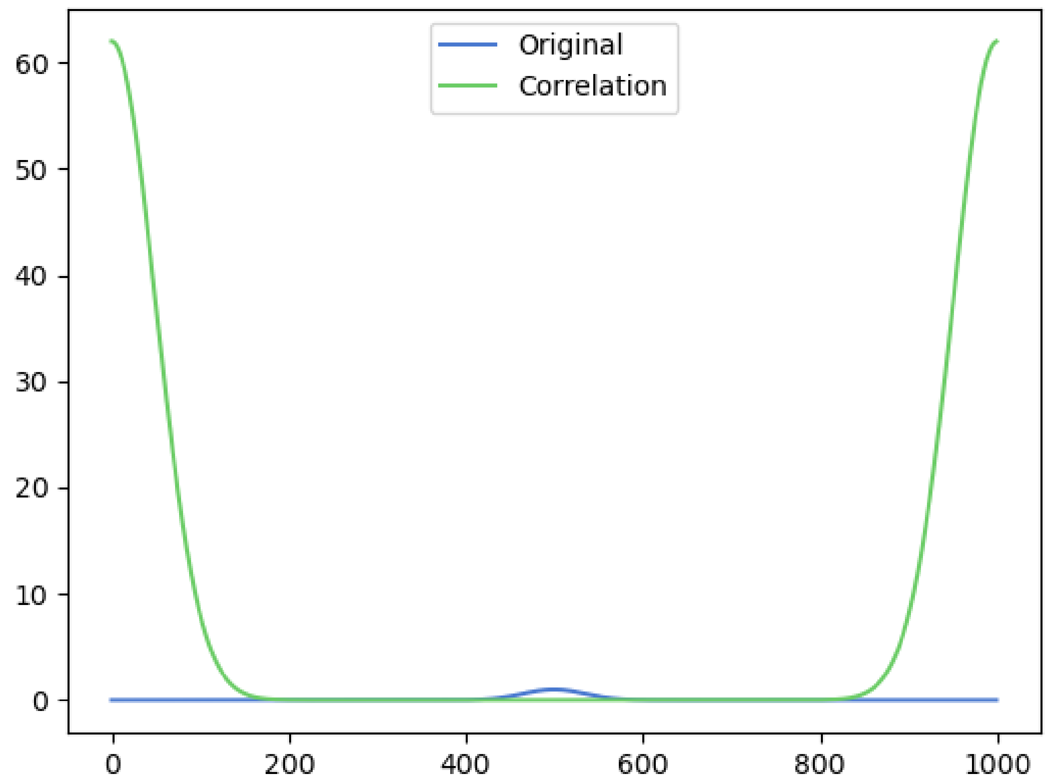

Figure 2: Correlation function of a Gaussian with itself. Result is as expected, where the correlation function is a Gaussian centered at 0 (ignoring wrap around), implying that the two functions are aligned perfectly (which they are, by construction).

Problem 3

See `corr_vs_shifted_forward.mov` for an animation displaying the relationship between the correlation of the shifted Gaussian with the original Gaussian. As expected, the correlation function provides information on how many frames you must shift in order to align the original and shifted function. It also moves in the same direction as the shifted function. This tells us that in order to align the original function with the shifted function, we must shift the original function forward by the frames specified by the correlation.

Problem 4

See `conv_safe_example.mov` for an example of my safe convolution. Essentially, I padded the input with arrays with as many zeros as the max length of the input arrays. If one array was smaller, it got padded with an additional set of zeros to allow the two arrays to be of the same length. In the example provided by the animation, the input Gaussian was an array of length 1000. I convolved this with a delta function of length 2000, where it iterated through shifts from 0 frames to 2000 frames. The result was an output array of length 4000 where the input Gaussian had been shifted by 2000 frames without wrapping around (as it would've in previous questions).

Problem 5

a)

Letting $\alpha = \exp(-2\pi ik/N)$, and using the definition of the geometric series that

$$\sum_{x=0}^{N} ar^x = a\frac{1-r^{N+1}}{1-r}, \tag{1}$$

we find that

$$\begin{aligned} \sum_{x=0}^{N-1} e^{-2\pi ikx/N} &= \sum_{x=0}^{N-1} \alpha^x \quad (2)\\ &= \frac{1-\alpha^N}{1-\alpha} \quad (3)\\ &= \frac{1-e^{-2\pi ik}}{1-e^{-2\pi ik/N}} \quad \square \quad (4) \end{aligned}$$

b)

$$\begin{aligned} \lim_{k\longrightarrow 0} \frac{1-e^{-2\pi ik}}{1-e^{-2\pi ik/N}} &= \textit{L'Hopitals Rule}\\ &= \lim_{k\longrightarrow 0} \frac{2\pi i e^{-2\pi ik}}{2\pi i e^{-2\pi ik/N}/N}\\ &= \boxed{N} \end{aligned}$$

Notice that when k is an integer but not a multiple of N, $-2\pi k/N$ will never be an integer multiple of 2π. Consequently, $e^{-2\pi ik/N}$ will never equal 1. And so the denominator of the following equation will never be zero, and thus

$$\frac{1-e^{-2\pi ik}}{1-e^{-2\pi ik/N}} = \frac{1-1}{c} = \boxed{0}, \tag{5}$$

where c is some nonzero constant.

c)

Applying the definition of the DFT, we find that for some non-integer k:

$$\begin{aligned}
f(k') &= \sum_{x=0}^{N-1} f(x) e^{-2\pi ik'x/N} && (6)\\
&= \sum_{x=0}^{N-1} \sin(2\pi kx/N) e^{-2\pi ik'x/N} && (7)\\
&= \sum_{x=0}^{N-1} \frac{e^{2\pi ikx/N} - e^{-2\pi ikx/N}}{2i} e^{-2\pi ik'x/N} && (8)\\
&= \frac{1}{2i}\sum_{x=0}^{N-1} e^{-2\pi i(k'-k)x/N} - e^{-2\pi i(k'+k)x/N} && (9)\\
&= \frac{1}{2i}\left(\frac{1-e^{-2\pi i(k'-k)}}{1-e^{-2\pi i(k'-k)/N}} - \frac{1-e^{-2\pi i(k'+k)}}{1-e^{-2\pi i(k'+k)/N}}\right) && (10)
\end{aligned}$$

I decided to only consider the real fourier transform since the input is purely real. Consequently, the results only consider the first $N//2+1$ data points. Comparing the real analytic fourier transform to the rFFT yields the following results:

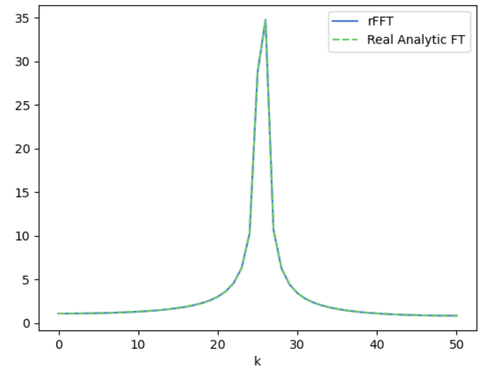

Figure 3: Comparison of analytic real Fourier transform and numpy's rFFT for a sine wave with $k = 25.55$.

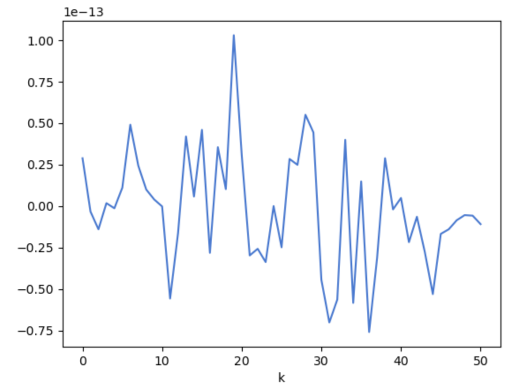

Figure 4: Residual plot. Absolute value of standard deviation of the above plot quoted an error of $\sigma = 3.5 \times 10^{-14}$.

As discussed in the figure, the error between the analytic FT and FFT is on the order of 10^{-14}. Although the DFT of a pure sine wave is a delta function analytically, the numerical result suffers since it is limited in it's resolution, especially around the k value. One could increase the number of points to increase the precision, but there will always be a set of non-zero amplitudes for k' not equal to k.

d)

By introducing a window function, we take care of the spectral leakage as seen in the following figure:

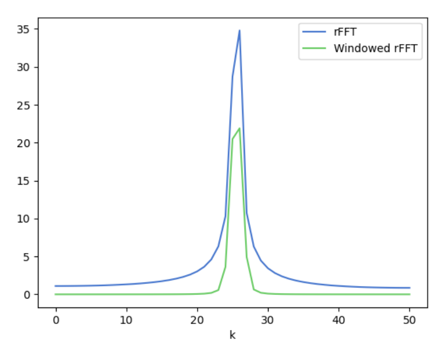

Figure 5: Comparison of windowed rFFT and non-windowed rFFT. Notice that the windowed rFFT's amplitude has been decreased, but the spectral leakage has been taken care of.

e)

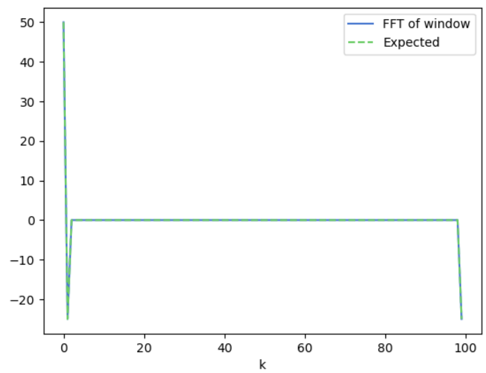

Figure 6: Comparison of the FFT of the window function versus what is expected as written by Jon in the assignment.

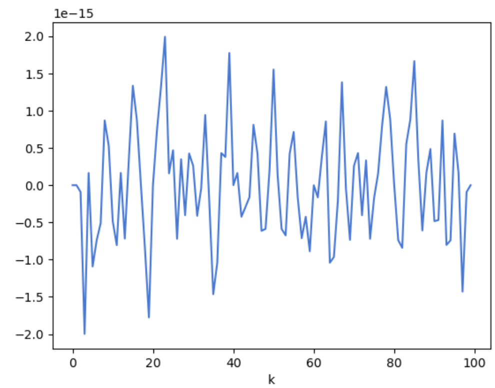

Figure 7: Residual plot highlighting that they agree with each other to within machine precision.

Lastly, following a similar procedure as done in class, we can obtain the windowed fourier transform by combining the weights. The results are given below:

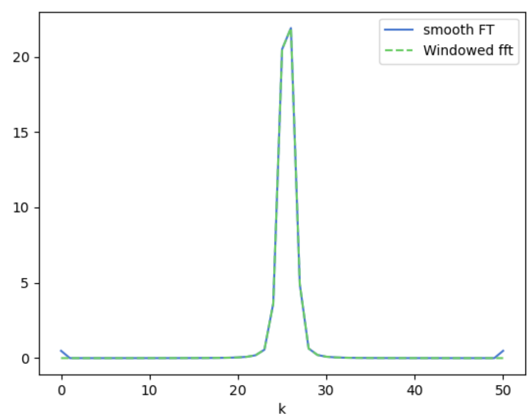

Figure 8: Comparison of windowed FFT versus smooth FT obtained by combining the weights and using `np.roll`. They agree with one another except for at the end points.

Problem Set 5

Problem 6

a)

In class, we ended the derivation at the following step:

$$\langle f(x)f(x+dx)\rangle = N - \frac{nx}{2} = g(x) \tag{11}$$

To get the power spectrum, we compute the fourier transform:

$$\begin{aligned} G(k) &= \int e^{2\pi ikx/N} g(x)dx \qquad (12) \\ &= \int_{-\infty}^{\infty} e^{2\pi ikx/N}\left(N - \frac{nx}{2}\right)dx \qquad (13) \\ &= \frac{(2\pi inNkx - N^2n - 4\pi iN^2k)e^{2\pi ikx/N}}{8\pi^2k^2}\Big|_{x=-\infty}^{x=\infty} \qquad (14) \\ &\propto \boxed{\frac{1}{k^2}} \qquad (15) \end{aligned}$$

where I employed the use of an integral calculator to calculate the above integral.

b)

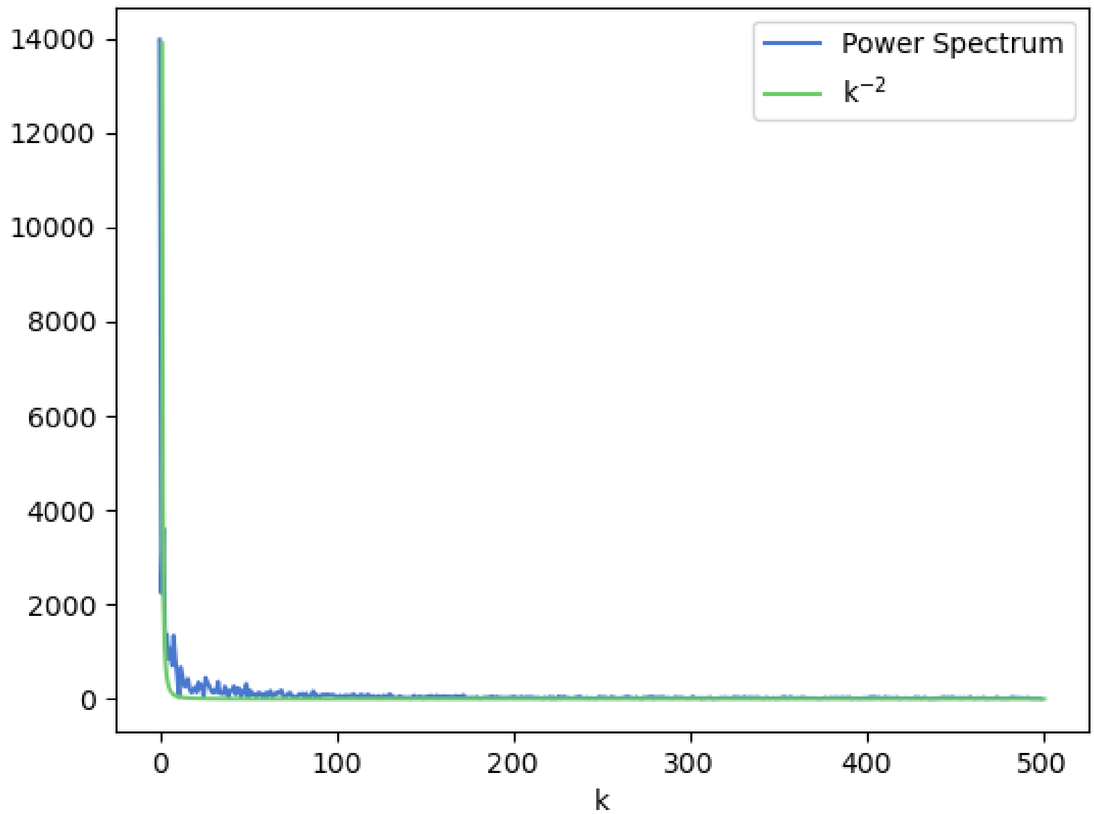

Figure 9: Comparison of power spectrum and $5N(k-0.7)^{-2}$, showing that the power spectrum indeed goes like k^{-2}.