PS6 : Phys 512

1) A convolution has the following form in both the continuous & discrete limit:

$$h(y) = \int dx\, f(x)\, g(y-x) \iff h_i = \sum_j f_j\, g_{i-j}$$

We want to shift the elements of an array so we say $h_i = f_i'$ is the shifted array & we have:

$$f_i' = \sum_j f_j\, g_{i-j}$$

Here we see the purpose of g_{i-j} is to shift the elements of f_i but how do we construct g?

↳ For a shift by γ, the new matrix should have the elements $f_i' = f_{i-\gamma}$

↳ To see how this imposes a condition on g, let's expand the sum:

$$f_i' = f_0 g_{i-0} + f_1 g_{i-1} + f_2 g_{i-2} + \dots + \overbrace{f_{i-\gamma}\, g_{i-(i-\gamma)}} + \dots + f_\gamma g_{i-\gamma} + \dots$$

Thus to have $f_i' = f_{i-\gamma}$, we look at the term $f_{i-\gamma} g_{i-(i-\gamma)} =$ $f_{i-\gamma} g_\gamma$

↳ Naturally we set $g_\gamma = 1$ & $g_j = 0$ for $j \neq \gamma$ & thus for a shift γ, the new element is simply:

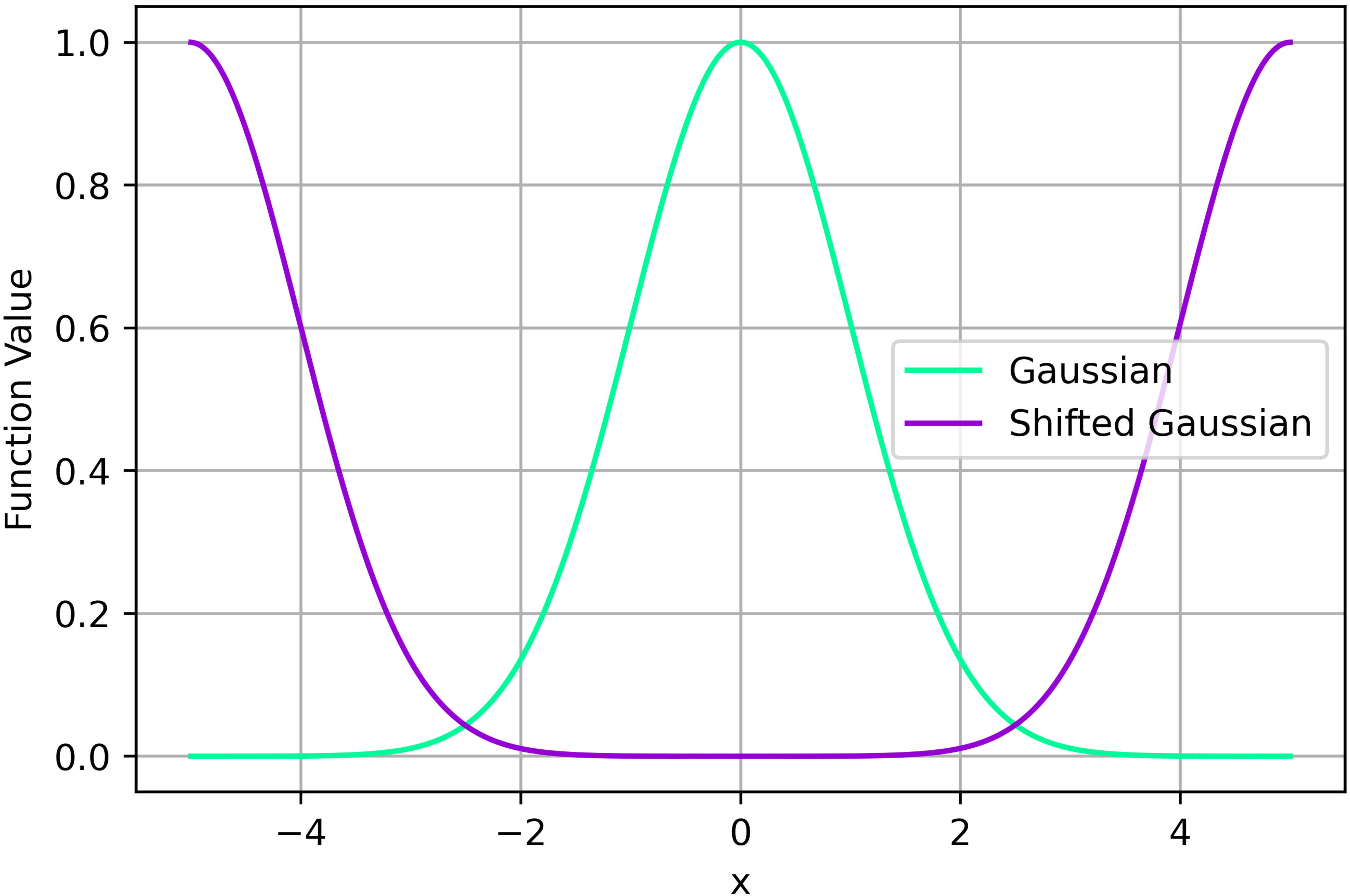
$$f_i' = \sum_j f_j g_{i-j} \longrightarrow f_{i-\gamma} g_\gamma$$

Thus our routine will have $g_j = \begin{cases} 1, & j = \gamma \\ 0, & j \neq \gamma \end{cases}$ as the shifting array & we shift f_j via the convolution $f' = f * g$

↳ Moreover, to compute this we make use of the convolution theorem:

$$f' = f * g = F^{-1}[\hat{f}\hat{g}] \text{ w/ } \hat{f} \equiv F[f]$$

The Gaussian plot looks like the following:

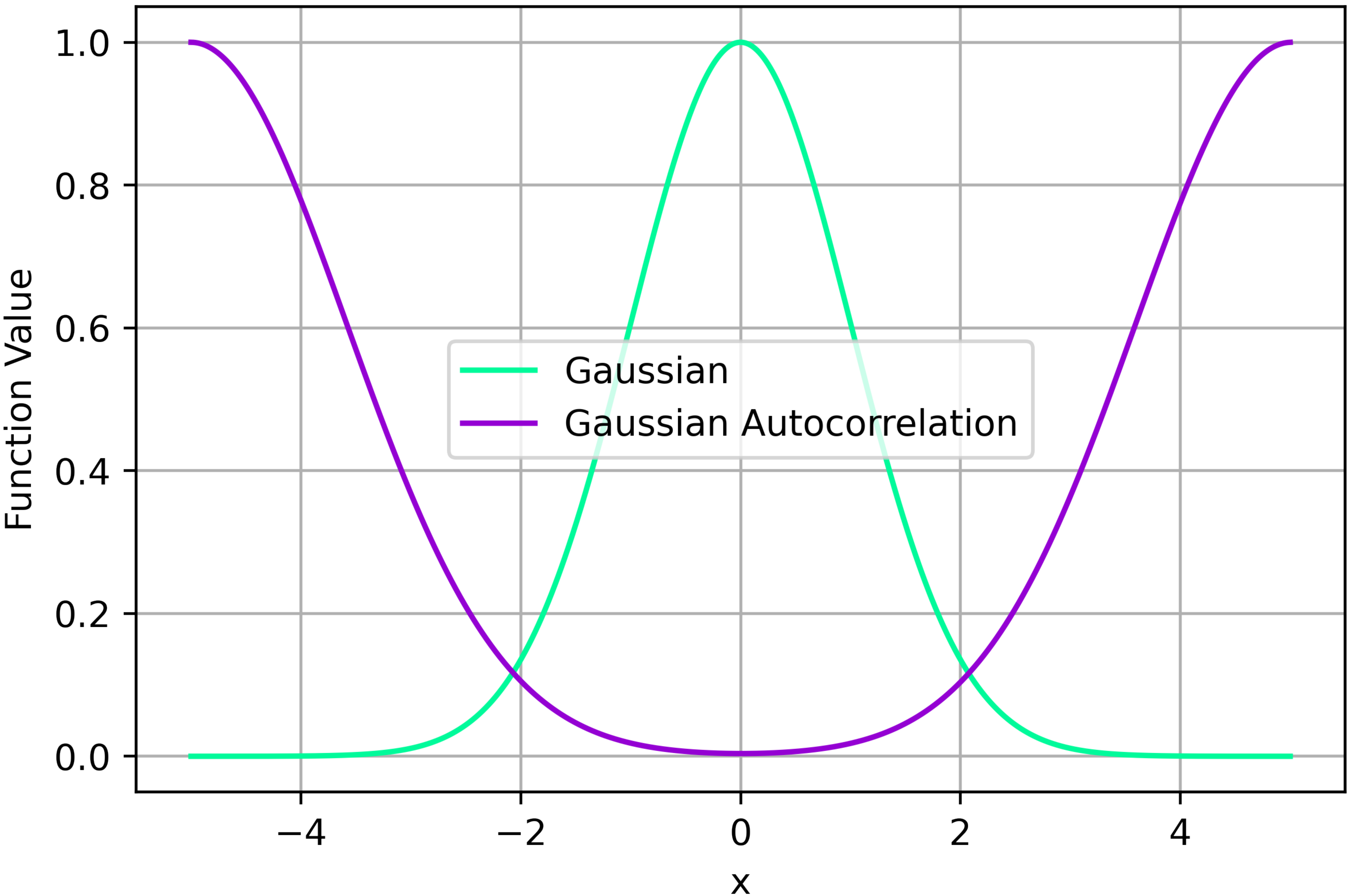


1.0
0.8
0.6
0.4
0.2
0.0
Function Value
−4
−2
0
2
4
x
Gaussian
Shifted Gaussian

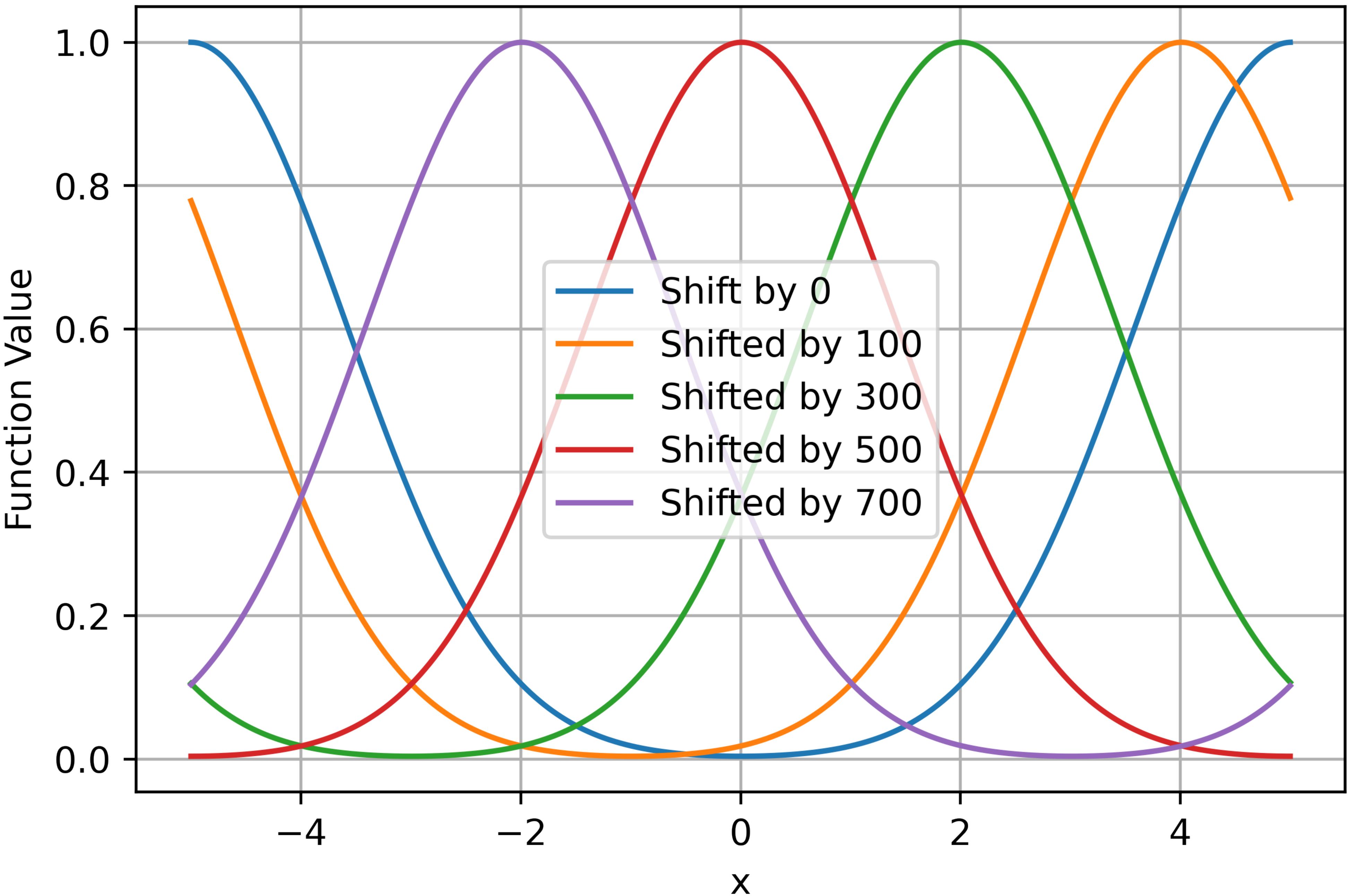
2a) Now we make use of another Fourier's representation analogous for the cross correlation of two functions:

$$f' = f \otimes g = F^{-1}[\hat{f}\hat{g}^{+}] \quad w/ \quad \hat{g}^{+} = (\hat{g}^{T})^{*}$$

Using this, the correlation of the Gaussian looks like: (A


Gaussian
Gaussian Autocorrelation
Function Value
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4
x

b) We shift the Gaussian by different arbitrary amounts to correlate those w/ the original Gaussian(s) to get the following plot:



Shift by 0
Shifted by 100
Shifted by 300
Shifted by 500
Shifted by 700
Function Value
x
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4

The correlation function depends on the shift linearly; it simply shifts the result of the correlation by the shift

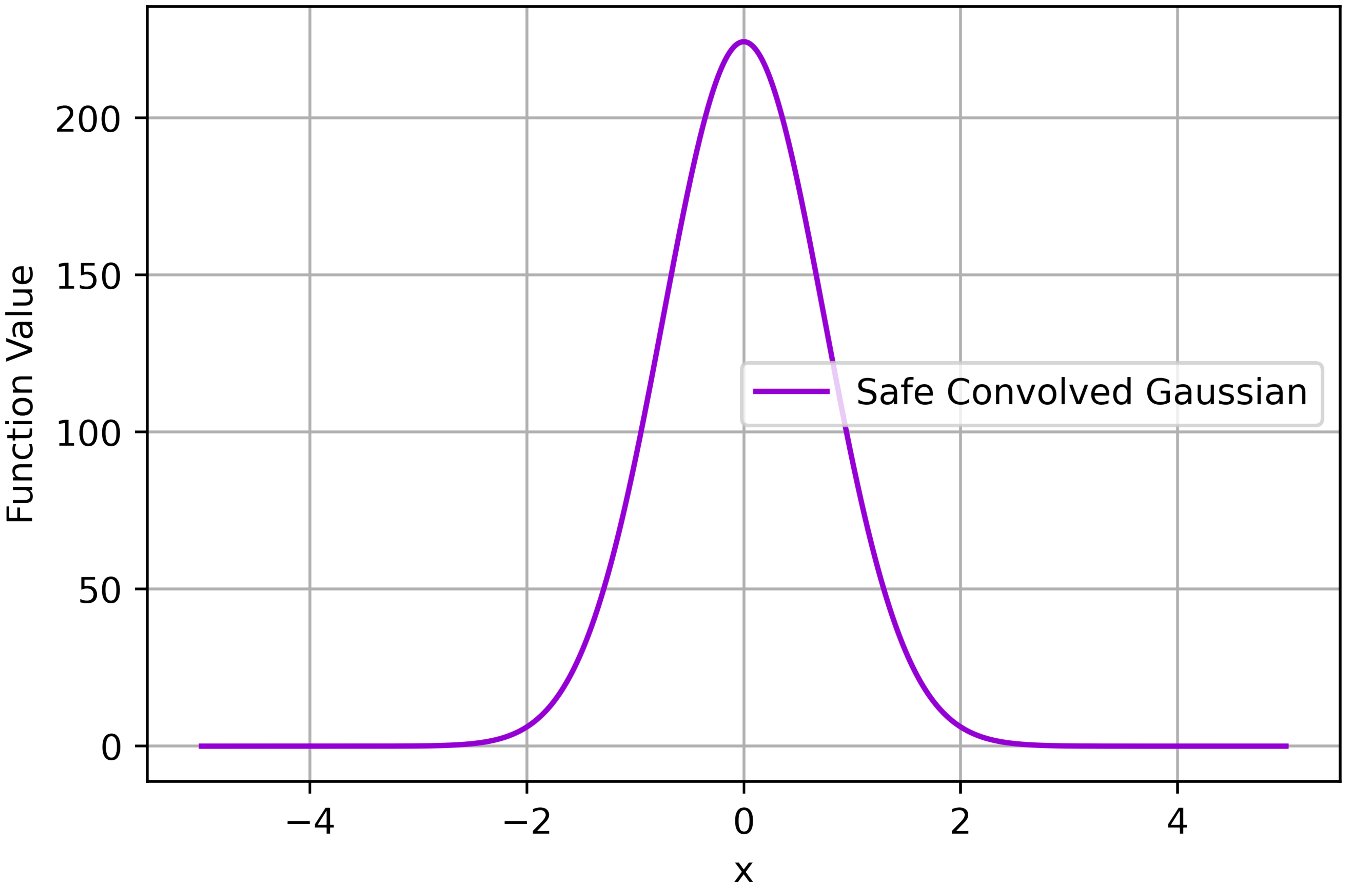
↳ This is not surprising (as hinted by Q3) of the circulant / wrap-around nature of the DFT

3) Now we wish to write a routine that avoids the circulant nature of the DFT

↳ This problem arises usually when the length of the arrays mismatch and the difference gets wrapped around via modulo

↳ To avoid this problem we pad the smaller array w/ zeros, where the amount of zeros padded is given by bigger Array − smaller Array

↳ The following is a plot of a convolution between two identical Gaussians w/ one being 1000 elements longer than the other:


Function Value
200
150
100
50
0
−4
−2
0
2
4
x
Safe Convolved Gaussian

4) a) $\gamma \equiv \sum_{x=0}^{N-1} \exp(-2\pi i k x/N) = \sum_{x=0}^{N-1} \alpha^x$ whe $\alpha = \exp(-2\pi i k/N)$

$$\rightarrow \gamma = \sum_{x=0}^{N-1} \alpha^x = \frac{1-\alpha^N}{1-\alpha} = \frac{1-\exp(-2\pi i k/\cancel{N})^{\cancel{N}}}{1-\exp(-2\pi i k/N)} = \frac{1-\exp(-2\pi i k)}{1-\exp(-2\pi i k/N)}$$

b) $\lim_{k\to 0} \gamma = \lim_{k\to 0} \frac{1-\exp(-2\pi i k)}{1-\exp(-2\pi i k/N)} = \frac{1-1}{1-1} = 0/0$ } Not well defined

We make use of L'Hopitals rule wrt k:

$$\lim_{k \to 0} \gamma = \lim_{k \to 0} \frac{\partial_k \left(1 - e^{-2\pi i k}\right)}{\partial_k \left(1 - e^{-2\pi i k/N}\right)} = \left. \frac{-(-2\pi i)\, e^{-2\pi i k}}{-(-2\pi i/N)\, e^{-2\pi i k/N}} \right|_{k=0}$$

$$= \frac{2\pi i}{2\pi i/N} = N$$

Moreover if $n, k \in \mathbb{Z}$ such that $k \neq nN$, then:

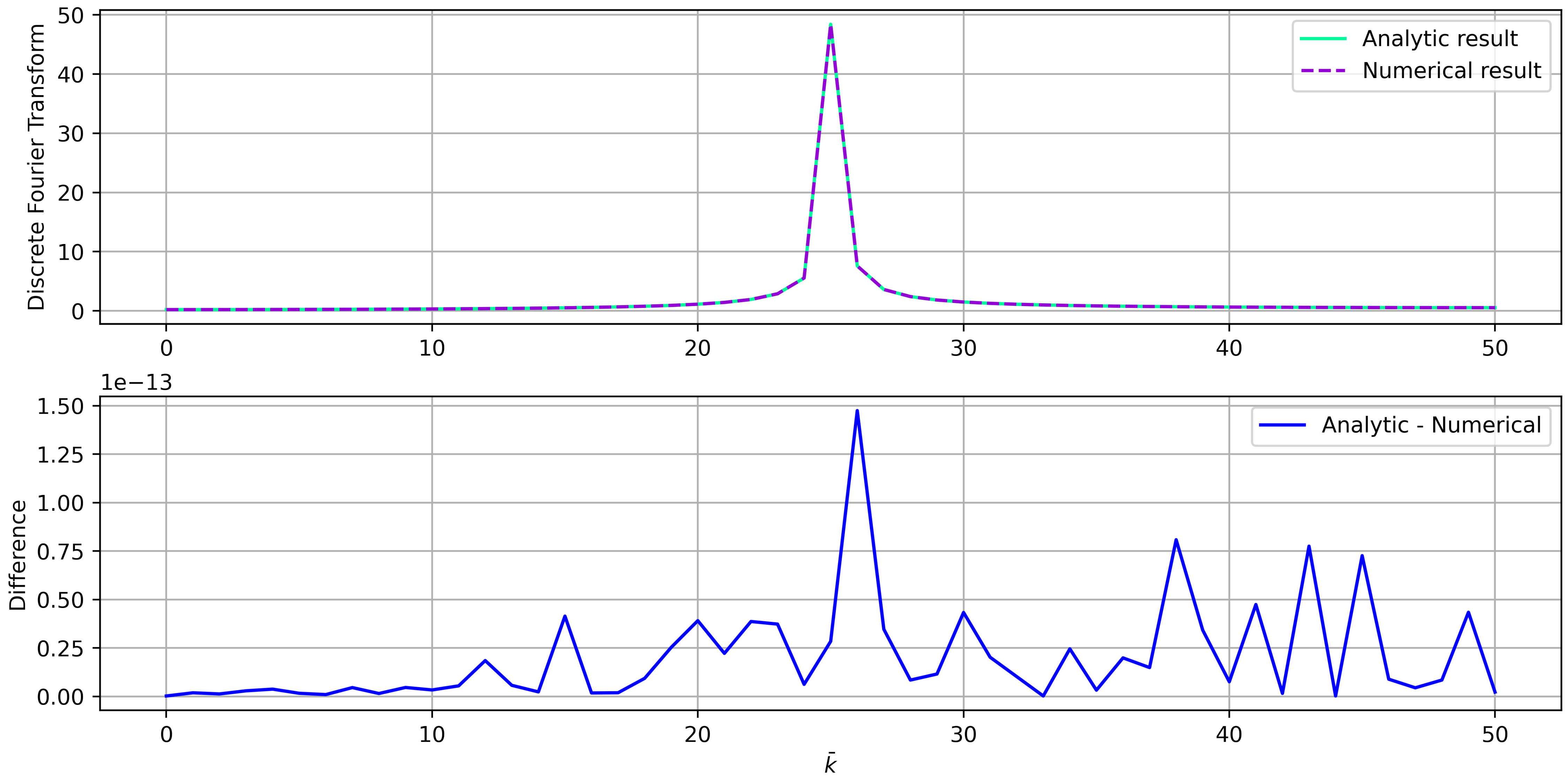
$$\gamma = \frac{1 - \cancel{e^{-2\pi i k}}^{\to 1}}{\underbrace{1 - \underbrace{e^{-2\pi i k/N}}_{\neq 1}}_{\neq 0}} = \frac{1-1}{1-e^{-2\pi i k/N}} = 0$$

c) I spent a while computing DFT[$\sin(2\pi k x)$] analytically & using the tricks from Q4a-b but although the fit looked close, the difference was of order $O(10^0) - O(10^1)$

↳ If you're curious you can look at the files in the 'old' folder!

We instead will compute the DFT somewhat analytically by doing the DFT sum $\sum_{x=0}^{N-1}$ with a few for loops & compare w/ np.fft

↳ We plot our 'analytic sum estimate' w/ the python FFT & their difference for $k = 8\pi$, $N = 100$:


Discrete Fourier Transform
Analytic result
Numerical result
0
10
20
30
40
50
1e−13
Difference
Analytic - Numerical
0.00
0.25
0.50
0.75
1.00
1.25
1.50
$\bar{k}$

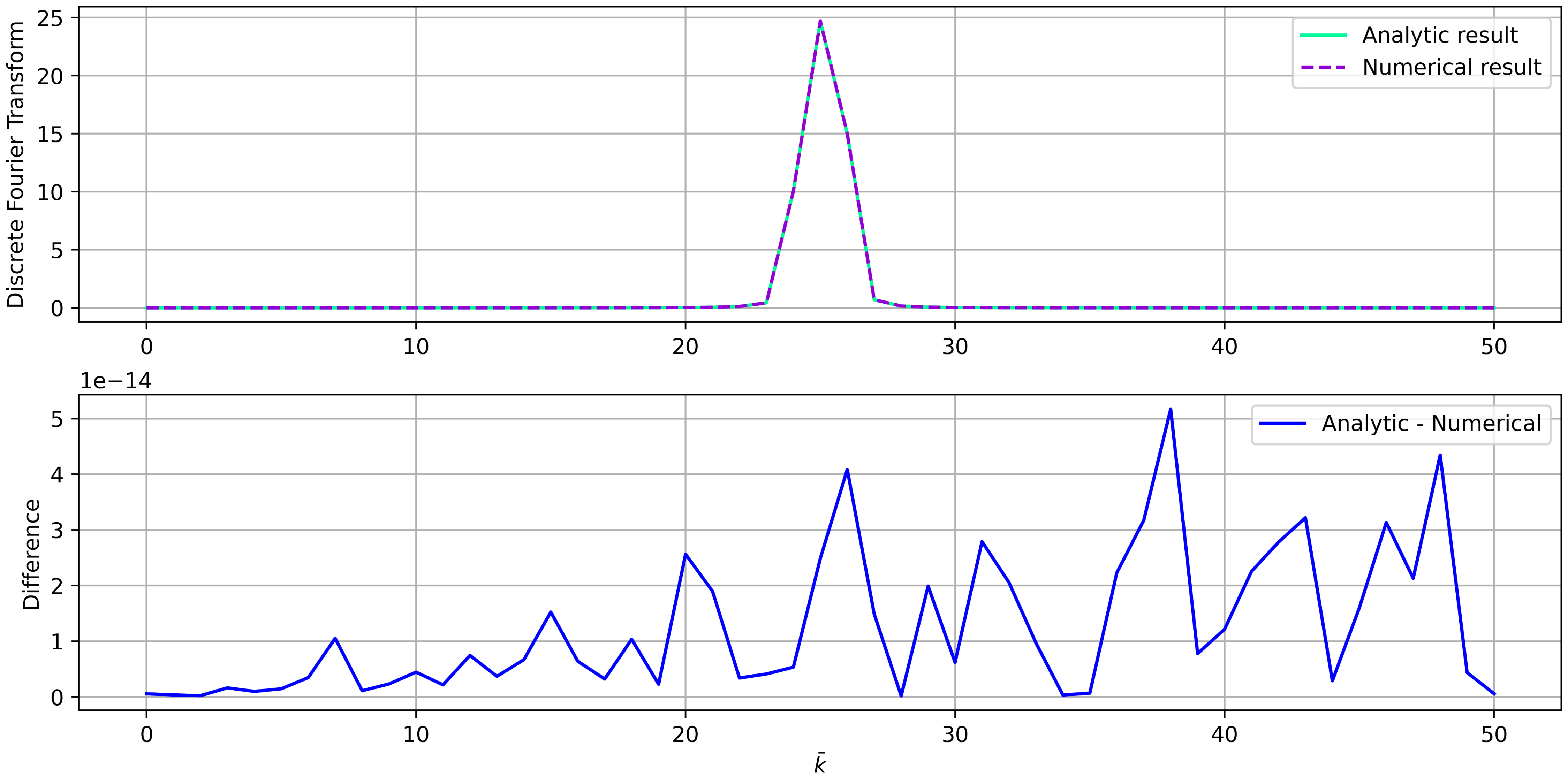
We can see they agree up to an order of magnitude $O(10^{-13})$. Note we plotted the modulus of the complex output.

↳ This looks too rigid (not smooth) to be a perfect delta function but we are close!

d) To prevent the leakage of modes at the edges of the interval, we make use of a window function w/ decaying edges

↳ A multiplication in position space is a convolution in Fourier space

The result of windowing the input function is seen in the following:



Discrete Fourier Transform
Analytic result
Numerical result
1e−14
Difference
Analytic - Numerical
$\bar{k}$

Although the error is better ($O(10^{-14})$), the peak seems to be wider & thus deviates more from a delta. I'm not sure about this result.

e) Let's Fourier transform the window function:

$$\overline{T} \equiv \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} dx\, e^{ikx} \left[\tfrac{1}{2} - \tfrac{1}{2}\cos(2\pi x/N) \right]$$

$$= \frac{1}{\sqrt{8\pi}} \left[\underbrace{\int_{\mathbb{R}} dx\, e^{ikx}}_{\gamma_1} - \underbrace{\int_{\mathbb{R}} dx\, e^{ikx} \cos(2\pi x/N)}_{\gamma_2} \right]$$

$\underline{\gamma_1}$: $\gamma_1 = \delta(k)$

$\underline{\gamma_2}$: $$\gamma_2 = \frac{1}{2} \int_{\mathbb{R}} dx\, e^{ikx} \left[e^{i2\pi x/N} + e^{-i2\pi x/N} \right]$$

$$= \frac{1}{2} \left[\int_{\mathbb{R}} dx\, e^{i(k+2\pi/N)x} + \int_{\mathbb{R}} dx\, e^{i(k-2\pi x/N)} \right]$$

$$= \tfrac{1}{2} \left[\delta(k+2\pi/N) + \delta(k-2\pi/N) \right]$$

$$\Rightarrow \overline{T} = \frac{1}{\sqrt{8\pi}} \left[\delta(k) + \tfrac{1}{2}\delta(k+2\pi/N) + \tfrac{1}{2}\delta(k-2\pi/N) \right]$$

We thus have 3 spikes at $N/2$, $N/4$ & $N/4$

$\Rightarrow$ The Fourier transform of the window is $[N/2 \;\; N/4 \;\; 0 \;\ldots\; 0 \;\; N/4]$

FT ≡ Fourier transform

Now, we wish to show that the Fourier'd window function is recoverable by some combination of each point of the un-windowed FT & some neighbours

↳ Let the unwindowed FT array be $F[i]$, $i = 0, \ldots, N-1$ & the FT of the window be $G[i] = [N/2, N/4, 0, \ldots, 0, N/4]$ the convolution is thus:

$$\gamma[i] = \sum_{j=0}^{N-1} F[j]\, G[i-j]$$

The only terms that survive are $i-j = 0, 1, -1$ thus $j = i, i+1, i-1$:

$$\gamma[i] = F[i]\, G[0] + F[i+1]\, G[1] + F[i-1]\, G[-1]$$

$$= \underbrace{\frac{N}{2} F[i] + \frac{N}{4} F[i+1] + \frac{N}{4} F[i-1]}$$

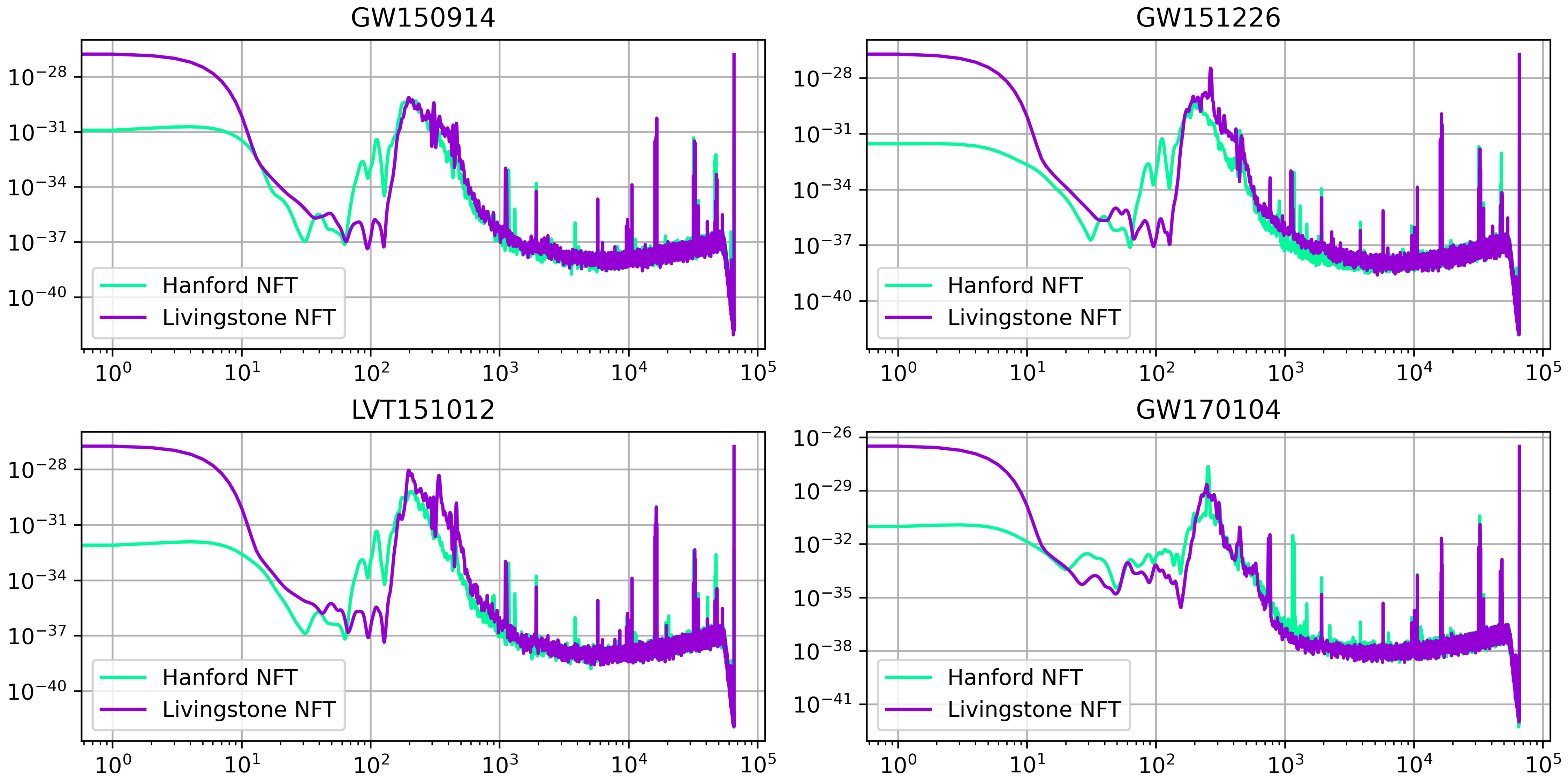
Windowed FT recovered via certain points of the unwindowed FT

5) a) Making use of the Wiener-Khinchin theorem ($Corr(g,g) = |F(k)|^2$), we model the noise as the power spectrum of the strain: $F[N] = |F[S]|^2$

noise ≡ NFT   strain ≡ SFT

Moreover, we smooth the data by averaging the noise w/ the flat rolling window function from class, over the two adjacent neighbours.

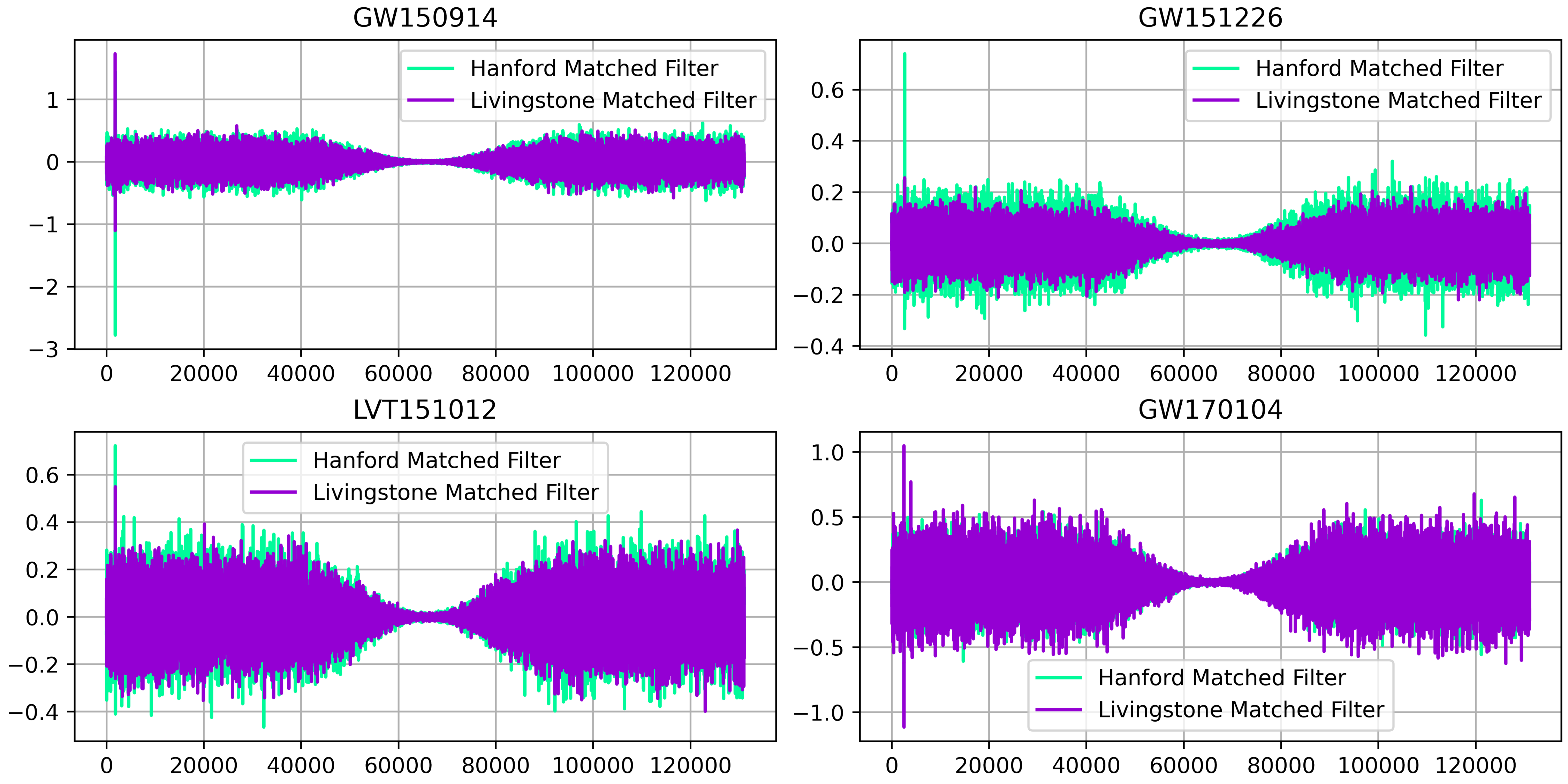
↳ The noise estimate for each event (NFT) can be seen in the following plot:


GW150914
GW151226
LVT151012
GW170104
Hanford NFT
Livingstone NFT
10^{-28}
10^{-31}
10^{-34}
10^{-37}
10^{-40}
10^{-26}
10^{-29}
10^{-32}
10^{-35}
10^{-38}
10^{-41}
10^0
10^1
10^2
10^3
10^4
10^5

b) Now we want to search the four sets of events for any signals lurking using a matched filter

↳ We do this by cross correlation the templates w/ the data which should include the signal & the noise, using FFTs

↳ The output of the match filters for each event looks like:


GW150914
Hanford Matched Filter
Livingstone Matched Filter
GW151226
Hanford Matched Filter
Livingstone Matched Filter
LVT151012
Hanford Matched Filter
Livingstone Matched Filter
GW170104
Hanford Matched Filter
Livingstone Matched Filter

Note: We prewhitened the strain & templates prior to the match filter.

We see the spikes on the left are the signals we are looking for in each plot/event.

c) Now we wish to compute the signal to noise ratio (SNR) by taking the quotient of the match filter output to the noise estimate of each event.

↳ How should we go about estimating the noise for each event? We take the std of the output of the match filter (making sure we work in the region in which the window function isn't suppressing the data) (we cut anything past 40,000 on the horizontal axis)

↳ Moreover, to get the SNR for the combined data of the detectors, we do:

$$SNR_c^2 = SNR_H^2 + SNR_L^2$$

The results of the SNR analysis are the following:

GW150914 : $SNR_H = 16.5964$, $SNR_L = 11.8401$, $SNR_c = 20.4160$

LVT151012 : $SNR_H = 6.4956$, $SNR_L = 5.4856$, $SNR_c = 8.5020$

GW151226 : $SNR_H = 9.8021$, $SNR_L = 4.7836$, $SNR_c = 10.9071$

GW170104 : $SNR_H = 7.3974$, $SNR_H = 6.4062$, $SNR_c = 9.7857$

d) First we define the signal template relative to the whitened noise model as TFT

↳ Then, the analytic form of the SNR is:

$$SNR = \max_{\mathbb{R}} \left\{ \frac{\text{Match Filter}}{std(F^{-1}[F[TFT]])} \right\}$$

Computing this using the data for each event results in:

GW150914: $SNR_H = 9.8841$, $SNR_L = 13.5327$, $SNR_c = 16.7580$

LVT151012: $SNR_H = 7.2475$, $SNR_L = 6.1777$, $SNR_c = 9.5613$

GW151226: $SNR_H = 10.9164$, $SNR_L = 5.4238$, $SNR_c = 12.1900$

GW170104: $SNR_H = 5.1512$, $SNR_L = 6.8138$, $SNR_c = 8.5419$

The results are the same order of magnitude but not quite the values we expected

↳ Any significant deviation is due to the noise being underestimated in the match filter.

e) We want to find the frequency from each event where half the weight comes from above & below that frequency, respectively.

↳ Normally we'd set the integral over the range to ½ to isolate for the frequency, but in our discrete case we do:

$$\frac{1}{2} = \frac{1}{N} \sum_{\text{data}} TFT(k)$$

isolate for k

normalization

Note: Similar to what we did in c), we take a branch cut at 6000 to avoid the TFT being suppressed by the window

Note: These frequencies tell us where in the noise curve where the peak signal is sitting (-Jon)

The results of solving the fixed, normalized sum gives us the frequencies: f

GW150914: $f_H = 101.2188$ Hz , $f_L = 115.3750$ Hz

LVT151012: $f_H = 76.3438$ Hz , $f_L = 95.9063$ Hz

GW151226: $f_H = 74.3750$ Hz , $f_L = 107.1250$ Hz

GW170104: $f_H = 93.3750$ Hz , $f_L = 75.4688$ Hz

f) Finally, we wish to localize the time of arrival of the signal & the relative positional uncertainty from the two detectors

For the time localization part, I base my work done on Prof Dale Gary's radio astronomy lectures (6) from the NJ Institute of Technology:

Taking the detector seperation to be $\Delta x \simeq 2\times10^6$ m ($\simeq 2\times10^3$ km), the angular position of the event signal is:

$$\theta = \arcsin\left(\frac{ct}{\Delta x}\right)$$

ct is the path difference

Δx detector difference

More over the uncertainty propegated in the postion is:

$$\sigma_\theta = \arcsin\left(\frac{c\,\sigma_t}{\Delta x}\right)$$

Thus using the above we can calculate how well we can localize the time of arrival:

$$\sigma_\theta = 2.9666^\circ$$

Moreover, we expect the typical positional uncertainty between the two detectors to be of the order $O(10^3)$ m.