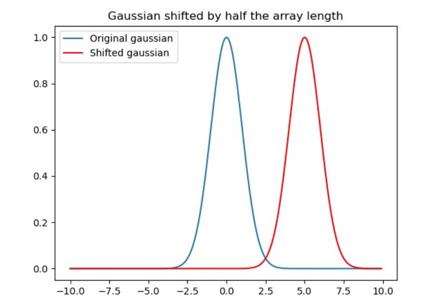
Assignment 6

1) Taking an array going from -10 to 10 with a step of 0.1 and a gaussian such that $f(x) = e^{-\frac{1}{2}x^2}$ we get:



2) a) We want to show that if $f * g = \int f(x)\, g(x+y)\, dx$

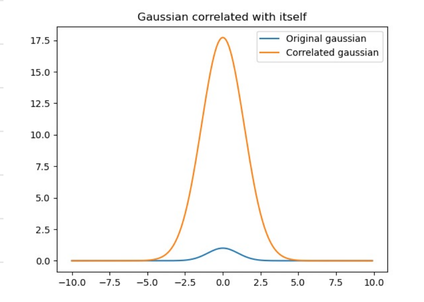
then $f * g = \text{ift}(\text{dft}(f) \times \text{conju}(\text{dft}(g)))$

We have
$$f * g = \int f(x)\, g(x+y)\, dx$$
$$= \Sigma_x \Sigma F(k)\, e^{2\pi i k x} \sum \text{conjugate}(G(k'))\, e^{2\pi i k' x}\, e^{2\pi i k' y/x}$$
$$f * g = \Sigma\Sigma\, F(k)\, \text{conj}(G(k'))\, e^{2\pi i k' y/N} \Sigma_x e^{2\pi i (k+k')x}$$

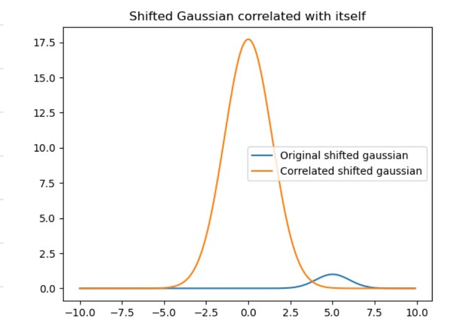
if $k' = -k$ then
$$= \Sigma\, F(k)\, \text{conj}(G(-k))\, e^{2\pi i k' y/N}$$
$$= \Sigma\, f(k)\, \text{conj}(G(-k))\, e^{-2\pi i k y/N}$$
$$= \Sigma\, f(k)\, \text{conj}(G(k))\, e^{2\pi i k y/N}$$
$$f * g = \text{ift}(\text{dft}(f) \times \text{conj}(\text{dft}(g)))$$

If $k' \neq -k$ then $f * g = 0$

We get for a gaussian correlated with its self



b) For the shifted gaussian correlated with it self we get:



The correlation does not depend on the shift. Namely we get the same result as for the non shifted gaussian.

We expected this because:

Let $f(x) = e^{-x^2}$ and $g(x) = f(x)$

Hence $f * g = \int e^{-x^2} e^{-(2x)^2} dx = \int e^{-x^2 - 4x^2} dx = \int e^{-5x^2} dx$

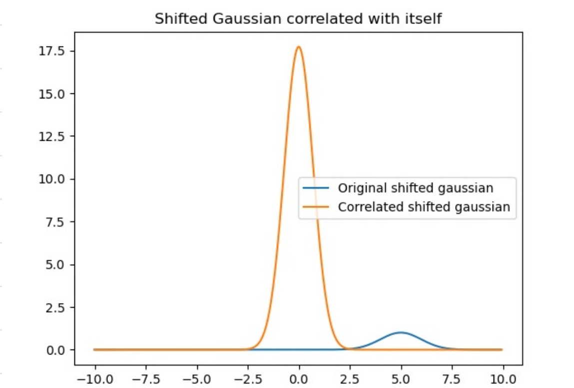
Let $f(x) = e^{-(x-dx)^2}$ and $g(x) = f(x)$

Hence $f * g = \int e^{-(x-dx)^2} e^{-(x-dx+x-dx)^2} dx = \int e^{-(x-dx)^2 - 4(x-dx)^2} dx = \int e^{-5(x-dx)^2} dx$

As $\int e^{-5x^2} dx = \int e^{-5(x-dx)^2} dx$ then the correlation of a shifted gaussian is the same as without the shift

3) Removing the FFT-based nature we get:

Adding 0's at the end of the array is supposed to remove the nature of the fft. We get:



4) a) $$\sum_{x=0}^{N-1} e^{-2\pi i k x/N} = \frac{1 - e^{-2\pi i k}}{1 - e^{-2\pi i k/N}}$$

We set set $\alpha = e^{-\frac{2\pi i k}{N}}$ starting from the LHS and using the geometric series formula

$$\sum_{x=0}^{N-1} \alpha^x = \frac{1 - \alpha^N}{1 - \alpha}$$

Plugging back $\sum_{x=0}^{N-1} e^{-2\pi i k x/N} = \dfrac{1 - \left(e^{-2\pi i k x/N}\right)^N}{1 - e^{-2\pi i k x/N}} = \dfrac{1 - e^{-2\pi i k}}{1 - e^{-2\pi i k/N}}$

b) let us show that as $k \to 0$ it equals N

we have $\lim_{k \to 0} \frac{1 - e^{-2\pi i k}}{1 - e^{-2\pi i k/N}} \overset{\text{L'Hopital's Rule}}{=} \lim_{k \to 0} \frac{2\pi i e^{-2\pi i k}}{\frac{2\pi i}{N} e^{-\frac{2\pi i k}{N}}} = \lim_{k \to 0} N e^{-2\pi i k + \frac{2\pi i k}{N}} = N$

let us show it is 0 if k is not a multiple of N

let $\forall k \exists l q / N = lk + q \iff$

Ergo
$$\frac{1 - e^{-2\pi i k}}{1 - e^{\frac{-2\pi i k}{lk+q}}}$$

In the numerator as $k \to 0$ we get 0

In the denominator as $k \to 0$ we don't get 0

c) We have the DFT of a sine wave such that

$$F(k) = \sum_{k=0}^{N-1} f(x) e^{\frac{-2\pi i k x}{N}} \quad \text{with } f(x) = \frac{1}{2i}\left(e^{ix} - e^{-ix}\right)$$

$$\text{thence } F(k) = \sum_{k=0}^{N-1} \frac{1}{2i} e^{ix\left(1 - \frac{2\pi k}{N}\right)} - \sum_{k=0}^{N-1} \frac{1}{2i} e^{-ix\left(1 + \frac{2\pi k}{N}\right)}$$

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

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

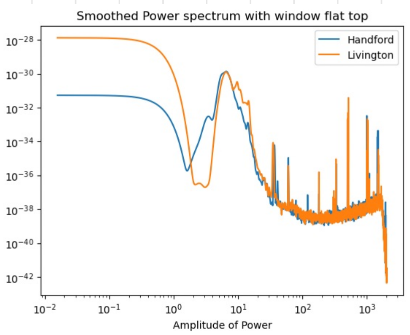
$$= \frac{1}{2i}\left(\left(1 - e^{-ix\left(1-\frac{2\pi k}{N}\right)} - e^{ix\left(1-\frac{2\pi k}{N}\right)}\right)^{-1}\left(1 - e^{ix(N-2\pi k)} - e^{-ix\left(1-\frac{2\pi k}{N}\right)} + e^{ix}\right)\right)$$

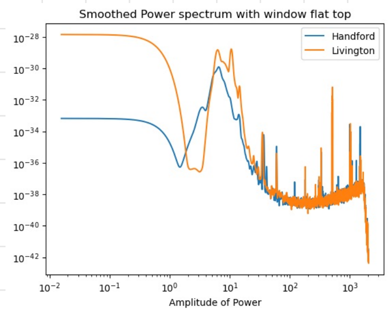
5) Matched filter of LIGO data

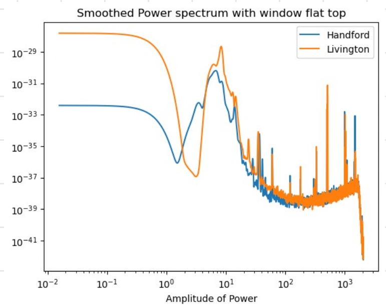
a) To get the noise model we:

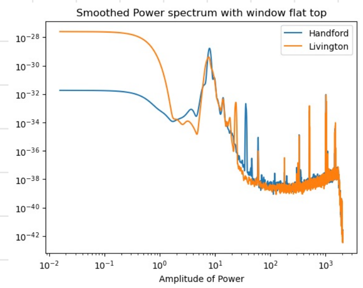
- extract the power spectrum from the data
- We smooth the power spectrum using the smooth_vector provided
- use a flat top window because we wanted an extend flat period near the center

From this procedure we get for the 4 events:

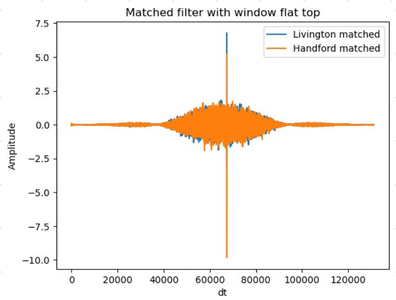


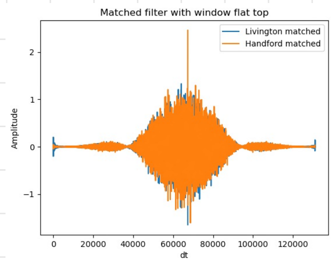


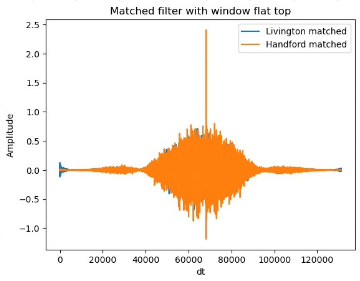


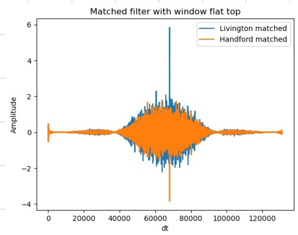


b) From noise model we manage to plot the 4 events with a matched filter









c) I was not able to do this part

I assume that it has to do with the covariance matrix that we have to express our noise model in some way that we could get for Livington and Handford data.

d)

e)

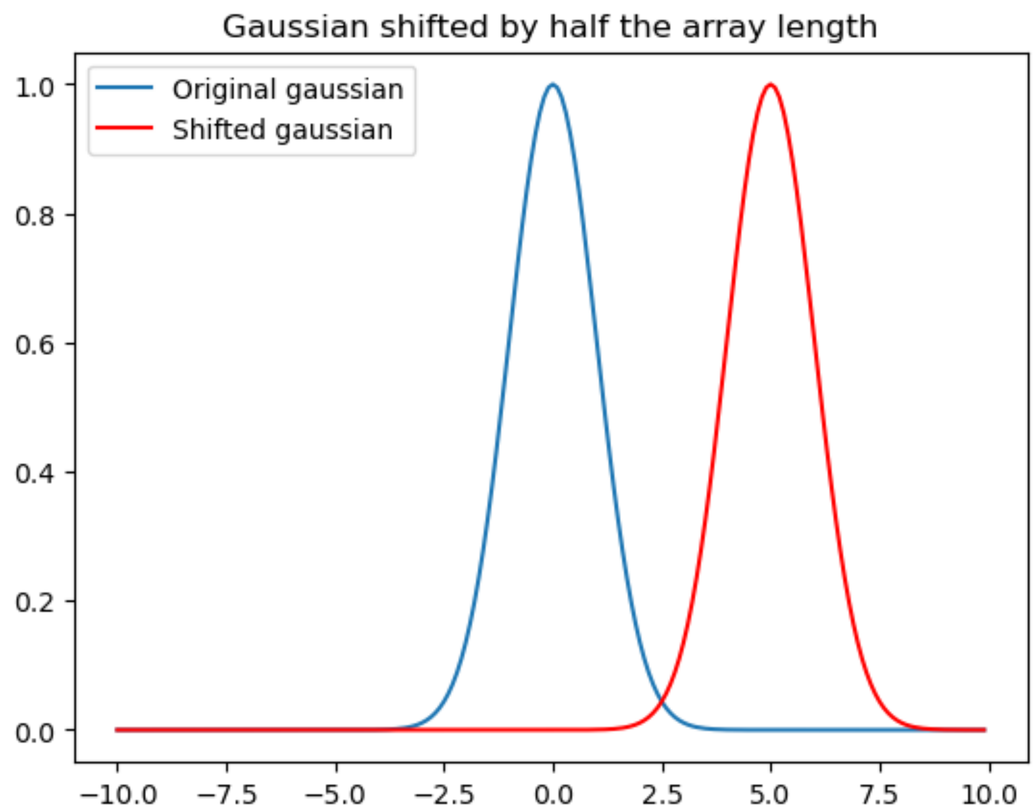
f)

code_pset6

November 12, 2022

1)

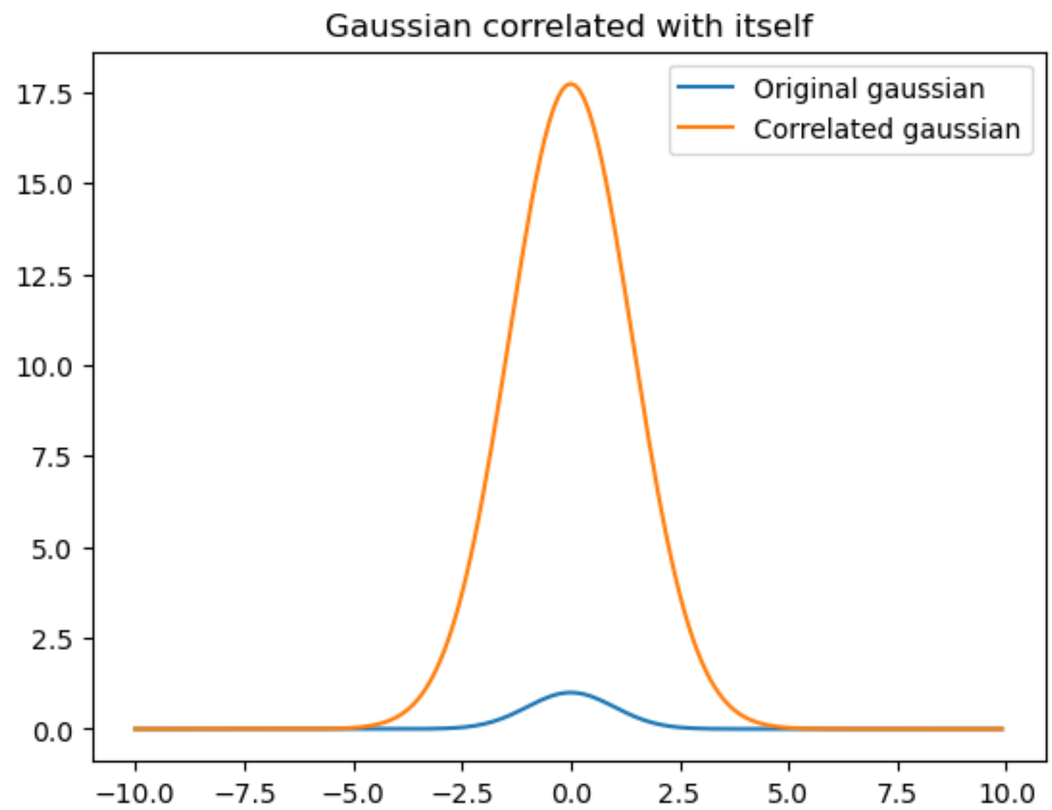
[1]:
```
import numpy as np
import matplotlib.pyplot as plt

#from the class code

def shift_conv(array, amount):
    y=np.exp(-0.5*array**2)
    N=array.size
    J=complex(0,1)
    yft=np.fft.fft(y)
    kvec=np.arange(N)
    dx=amount
    yft_shifted=yft*np.exp(-2*np.pi*J*kvec*dx/N)
    y_shifted=np.real(np.fft.ifft(yft_shifted))
    return y_shifted


x=np.arange(-10, 10, 0.1)
y=np.exp(-0.5*x**2)
y_shifted = shift_conv(x, x.size//4)#//2

plt.title("Gaussian shifted by half the array length")
plt.plot(x,y, label="Original gaussian")
plt.plot(x,y_shifted, 'r', label="Shifted gaussian")
plt.legend()
plt.show()
```



2) a)

[2]:
```python
import numpy as np
import matplotlib.pyplot as plt

#from class notes
def correlation(f,g):
    ft1 = np.fft.rfft(f)
    ft2 = np.conj(np.fft.rfft(g))
    return np.fft.irfft(ft1*ft2)



arr=np.arange(-10,10,0.1)
gauss=np.exp(-0.5*arr**2)

gauss_corr_itself=correlation(gauss,gauss)



plt.title("Gaussian correlated with itself")
```

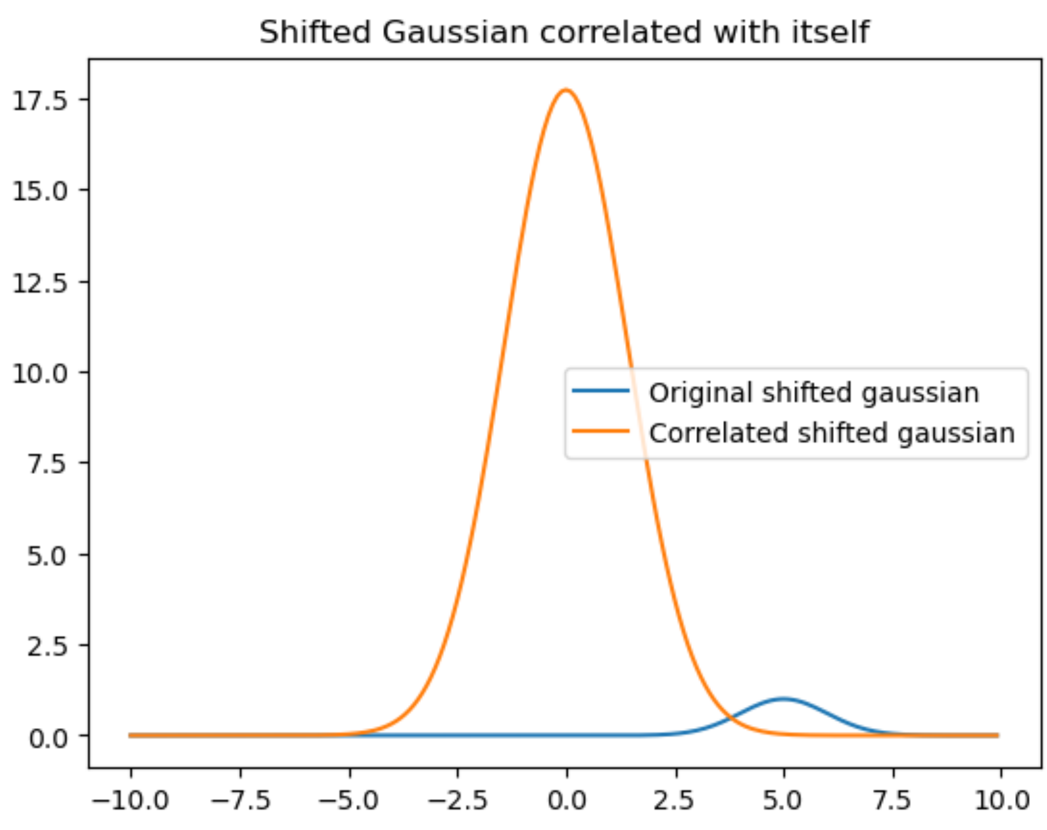
```python
plt.plot(arr, gauss, label="Original gaussian")
plt.plot(arr, np.fft.fftshift(gauss_corr_itself), label="Correlated gaussian")
plt.legend()
plt.show()
```



2) b)

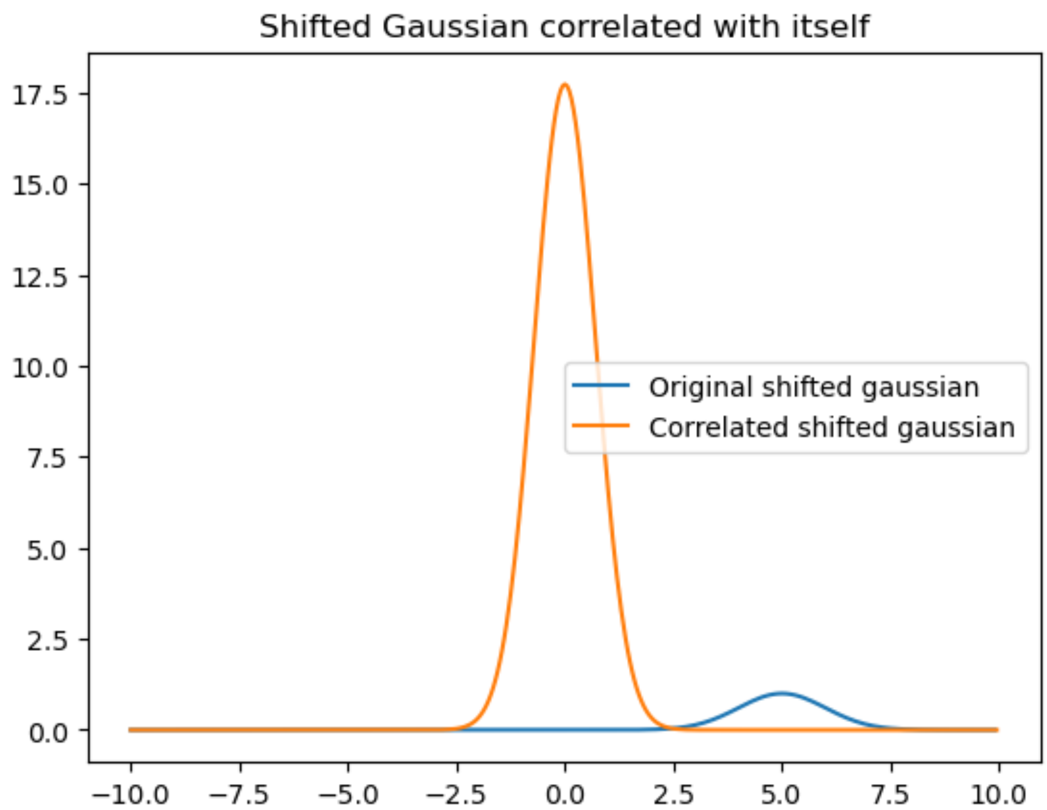
[3]:
```python
import numpy as np
import matplotlib.pyplot as plt

#from class notes
def correlation(f,g):
    ft1 = np.fft.fft(f)
    ft2 = np.conj(np.fft.fft(g))
    return np.real(np.fft.ifft(ft1*ft2))

def shift_conv(array, amount):
    y=np.exp(-0.5*array**2)
    N=array.size
    J=complex(0,1)
```

```
    yft=np.fft.fft(y)
    kvec=np.arange(N)
    dx=amount
    yft_shifted=yft*np.exp(-2*np.pi*J*kvec*dx/N)
    y_shifted=np.real(np.fft.ifft(yft_shifted))
    return y_shifted

arr=np.arange(-10,10,0.1)
gauss=np.exp(-0.5*arr**2)


gauss_shifted = shift_conv(arr, arr.size//4)#//2


gauss_corr_itself=correlation(gauss_shifted,gauss_shifted)

plt.title("Shifted Gaussian correlated with itself")

plt.plot(arr, gauss_shifted, label="Original shifted gaussian")
plt.plot(arr, np.fft.fftshift(gauss_corr_itself), label="Correlated shifted␣
↪gaussian")
plt.legend()
plt.show()
```



3)

[4]:
```
import numpy as np
import matplotlib.pyplot as plt

#from class notes
def correlation(f,g):
    ft1 = np.fft.fft(f)
    ft2 = np.conj(np.fft.fft(g))
    return np.real(np.fft.ifft(ft1*ft2))

def correlation_fixed(arr, arr1):
    for i in range(arr.size):
        arr = np.append(arr, np.array([0]))
        arr1 =np.append(arr1, np.array([0]))
    return correlation(arr, arr1)
def shift_conv(array, amount):
    y=np.exp(-0.5*array**2)
    N=array.size
    J=complex(0,1)
    yft=np.fft.fft(y)
    kvec=np.arange(N)
    dx=amount
    yft_shifted=yft*np.exp(-2*np.pi*J*kvec*dx/N)
    y_shifted=np.real(np.fft.ifft(yft_shifted))
    return y_shifted

arr=np.arange(-10,10,0.1)
gauss=np.exp(-0.5*arr**2)


gauss_shifted = shift_conv(arr, arr.size//4)#//2


gauss_corr_itself=correlation_fixed(gauss_shifted,gauss_shifted)
plt.title("Shifted Gaussian correlated with itself")

plt.plot(arr, gauss_shifted, label="Original shifted gaussian")
arr=np.arange(-10,10,0.05)
plt.plot(arr, np.fft.fftshift(gauss_corr_itself), label="Correlated shifted␣
↪gaussian")
plt.legend()
plt.show()
```


Shifted Gaussian correlated with itself
Original shifted gaussian
Correlated shifted gaussian
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0