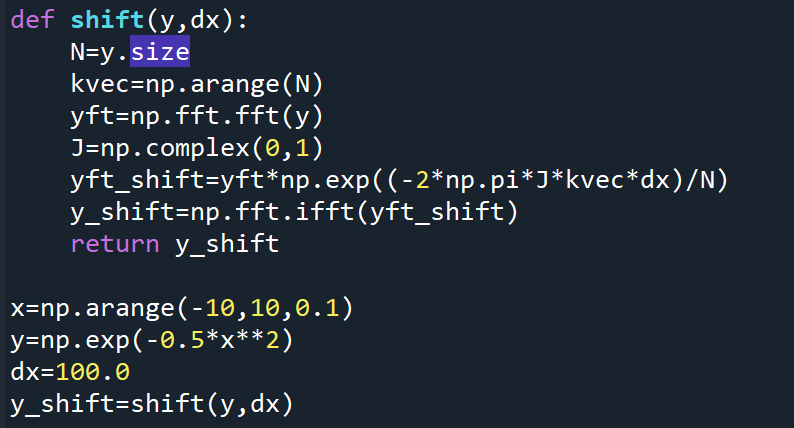
**Assignment 4**

**Yifei Gu**

**260906596**

**Q1)**

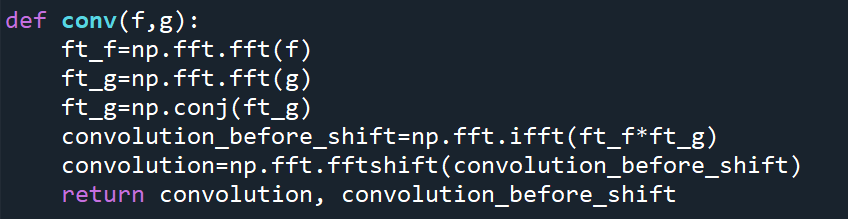
Here is my defined function and choice of gaussian function:

**Chart, histogram

Description automatically generated**

The gaussian is shifted by half of the array length.

**Q2)**



The result from **np.fft.fft** is not centered. Considering the periodicity, I used **np.fft.fftshift** to **move** the convolution **to the center**. Here I showed both plots.

**Chart

Description automatically generated**

The gaussian I chose is **y= exp(-0.5\*x\*\*2)**. This plot is not easy to interpret. However, when shift the convolution, which is a gaussian, to the center, it makes much more sense now. A **fat gaussian** becomes a **skinny gaussian** after Fourier Transform.

**Chart, line chart

Description automatically generated**

**Q3)**

**Chart

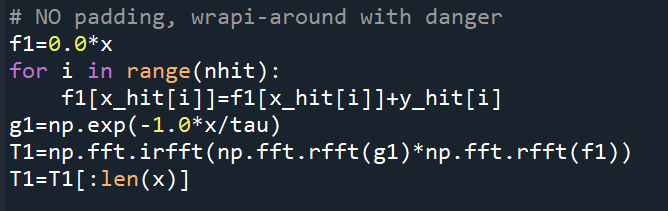
Description automatically generated**

Again, I used **np.fft.fftshift** to move the convolution to the center. Comparing with the convolution curve with guassian itself, the convolution curve between gaussian and its shifted gaussian **moves** towards the shifted gaussian.

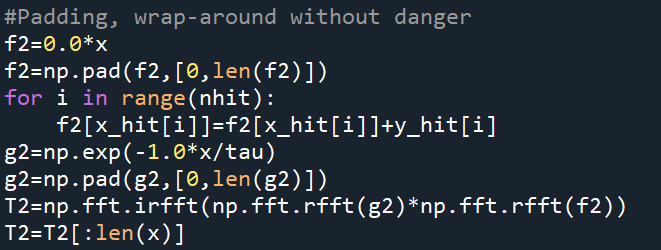
**Chart, line chart

Description automatically generated**

**Q4)**



Here I used the example showed in class. Doing convolution with no padding might cause the problem of wrap-around, which is due to the periodicity of Fourier transfer. As you can see from the plot, the curve without padding is trying to connect the starting point and ending point. **Padding the data with some zeros and chopping them off in the end will solve this problem.**



**Chart, line chart, histogram

Description automatically generated**

**Q5）**

**a)**

Let ,

Bring back to the equation:

**b)**

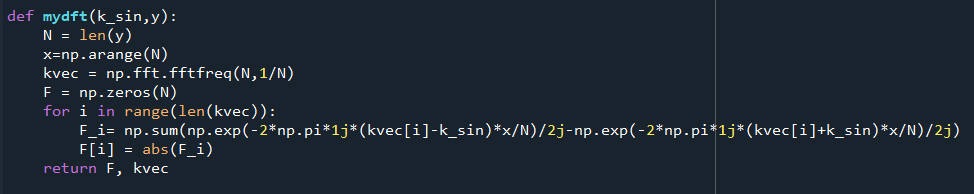
According to *L'Hôpital*’s rule, when k🡪0:

When integer k is not a multiple of N, k/N will not be an integer. , but . = 0.

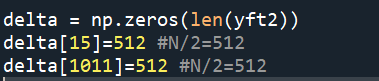
**c)**

According to **Euler's formula**:

With this **analytic estimate of the DFT**, I wrote my DFT function as below:



I chose **a non-integer k =15.4**, which will reflect as **a peak** in the Fourier Transform of the sine wave near **k’=15**. Due to aliasing, there be **another peak** **near k’=1009**, which is the **mirror image** about **N/2** of **k’=15**. **This is also consistent with the analytic estimate of DFT.** When **k’=k** or **k’=-k**, and are not zero and **all the other k’ will have a DFT equal to zero.** Here is how I defined my **delta function**:



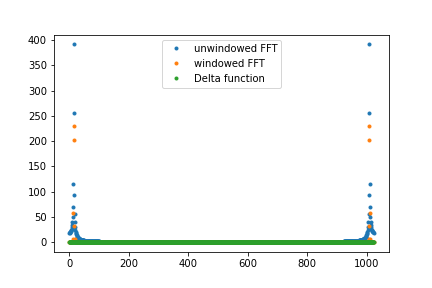
Chart

Description automatically generated

As you can see, the **numpy fft**(blue) is **overlapping** with “**my dft**”, but both have some **spectral leakage** from the delta function. The error **between my written DFT and numpy FFT** is close to the **machine precision**, and the error **between my written DFT and delta** function is **huge**:



**d)**



**With windowing** (Orange dots), the **spectral leakage** effects are **reduced**. The **error** between FFT and delta function also **dropped**.



**e)**

When k’=0, F(0)=N/2,

When k’=1, F(0)=-N/4,

When k’=-1, F(0)=-N/4,

All the other k’ will give a zero for DFT. The Fourier Transform of the window is [N/2, -N/4,0,0……0,-N/4].