Homework Assignment #2:

Due: End of Day Oct 24 (Friday)

Problem 1 Plot the real part, the imaginary part, and the length (or norm) of the following vectors in the Fourier Basis of $\mathbb{C}^{64}$:

$$F_0, F_1, F_{63}, F_2, F_{62}, F_8, F_{56}, F_{32}$$

Note that the FFT function in Matlab counts the indices from 1, 2, …, N instead of the usual CS programming convention 0, 1, 2, …, N-1. If you are using Matlab, remember to use the correct index.

Problem 2 Practice 1 Dimensional FFT and DCT.

Consider the following 512-dimension column vector z, where:

$$z(n) = \begin{cases} 0 & 0 \leq n \leq 64, \qquad 384 \leq n \leq 511 \\ 1 & \qquad\qquad 65 \leq n \leq 383 \end{cases}$$

Use Matlab or Python:

Calculate the Fourier Transform and Discrete Cosine Transform of z and plot the result. Is the Fourier Transform of z real? Explain why or why not.

Low Pass: For both the Fourier Transform and Discrete Cosine Transform, set the 128 entries of high frequency components to 0 and perform respective inverse transforms and plot the result.

High Pass: For both the Fourier Transform and Discrete Cosine Transform, set the 64 entries of low frequency components to 0 and perform respective inverse transforms and plot the result.

Compression: For both the Fourier Transform and the Discrete Cosine Transform, remove the 448 entries of the smallest norm or value and perform the respective inverse transforms and plot the result. Note that this is an 8:1 compression.

Problem 3 Prove Parseval's Theorem.

When **<u>normalized</u>** Fourier Transform is used, the Parseval's Theorem states that for $z, w \in \mathbb{C}^N$:

$$\langle z, w \rangle = \langle \hat{z}, \widehat{w} \rangle$$

where $\hat{z}$ and $\widehat{w}$ are the Fourier transforms of z and w respectively.

(Hint: this proof is very similar to the Plankerel's Theorem as presented in class.)

Problem 4 In the class, we discussed circular convolution:

$$(w * z)(m) = \sum_{n=0}^{N-1} w(m-n)z(n)$$

and the convolution theorem:

$$(w * z)^\wedge(m) = \hat{w}(m)\hat{z}(m)$$

The goal of this problem is to investigate the so-called "deconvolution" process. More specifically, assume that you are given the vectors z and x, can you find a vector w such that $w * z = x$? Develop a fast algorithm for calculating deconvolution.

Problem 5 Let w, z, and e_0 be N dimension complex vectors, and $e_0 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$.

We say that w is the convolution inverse of z if $w * z = e_0$.

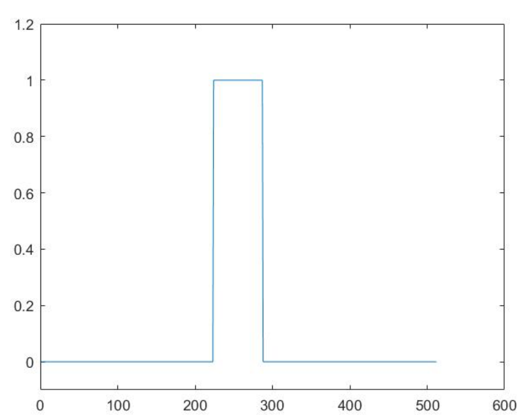
Design an efficient algorithm such that given a vector z, determine if z has a convolution inverse, and if yes find the convolution inverse of z.

Problem 6

In this problem, you will be exploring the frequency domain using both Fourier Transform and Discrete Cosine Transform.

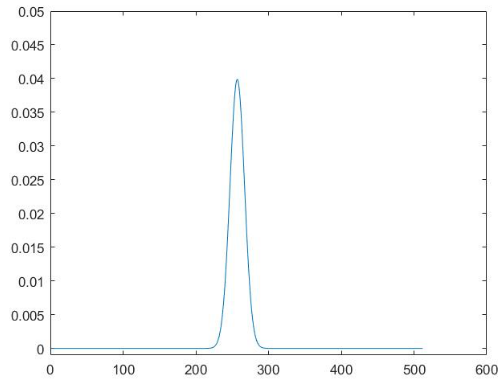
Suppose you are given the following target distribution:

$$D^*(n) = \begin{cases} 0 & 0 \le n \le 223, \quad\quad 288 \le n \le 511 \\ 1 & \quad\quad 224 \le n \le 287 \end{cases}$$



You may imagine that the given D^* is a radiosurgery dose prescription, which aims to deliver a high dose to the tumor and keeping the dose to everywhere else to 0.

Suppose that the Gamma Knife machine can deliver radiation "shots" of Gaussian dose distribution $N(\mu, 10)$. Notice that μ is the center of the distribution, while $\sigma = 10$ governs the shape of the distribution. You may imagine that a Gamma Knife Unit is capable of adjusting the μ to move the shots around. The figure below shows $N(256,10)$.




```
The 2 plots shown here are obtained using the following Matlab code:
D = zeros (1, 512);
D (1, 224:287) = 1 ;
figure(1) ;
plot (D) ;
ylim([-0.1, 1.2] );

x = 0:1:511;
y = normpdf(x, 256, 10) ;
figure(2) ;
plot(y) ;
ylim([-0.001, 0.05] );
```

The Gamma Knife Radiosurgery planning problem is to find the "beam-on" times t_j for a set of shots such that:

$$
\begin{aligned}
&min \quad \left\| \left(\sum_{j=224}^{287} t_j N(j, 10) \right) - D^* \right\|_2^2 \\
&s.t. \quad t_j \geq 0 \text{ for } j = 224, 225, \ldots, 287
\end{aligned}
$$

In this optimization, $\sum_{j=224}^{287} t_j N(j, 10)$ is the dose distribution created by the plan. The reason why $j = 224, 225, \ldots, 287$ is because it makes no sense to deliver any shots that is focused outside the tumor. The goal of the optimization is to minimize the difference $\left(\sum_{j=224}^{287} t_j N(j, 10)\right) - D^*$. The non-negative constraints are due to that beam-on times (i.e., weighting of the shots) must be non-negative.

Answer the following questions:

(1) Gain the basic understanding of the problem by solving the above non-negative least square optimization problem using Matlab or Python and plot $\sum_{j=224}^{287} t_j N(j, 10)$ vs D^*.

(The matlab routine for solving non-negative least square is lsqnonneg.)

(2) Perform the above optimization in the frequency domain using both Fourier Transform and Discrete Cosine Transform and compare the performance of the results with those from the spatial domain as in (1).

(3) One of issues of the above optimization is that the ideal dose distribution D^* has a perfectly sharp edge, which is impossible to obtain from Gaussian shots. To overcome this, one idea is to blur the sharp edges of the dose distribution D^* by removing its high frequency components that is not deliverable by a Gaussian distribution. Please implement this approach and see if it produces any improvement.