# Math 473 HW 3

Instructions: show all steps to get full credits. Assemble Matlab code and results and turn in together with written parts. No late homework is accepted in any case.

1. Prove that a 1D signal $x \in \mathbb{C}^N$ is real valued if and only if $\bar{X}_k = X_{-k}$ where $X_k$ is the discrete Fourier transform of $x$ corresponding to frequency $-N/2 \le k \le N/2 - 1$.

2. **Application of Fourier transform in Magnetic Resonance Imaging (MRI).** MRI is a widely used noninvasive medical imaging technique used in radiology to visualize detailed internal structure and limited function of the body. Raw MRI data, also known as k-space data is related to the image by 2D discrete Fourier transform. Let $I$ be an image, its k-space data is $K = fftshift(fft2(I))$ with 0 frequency in the center and high frequency away from the center.

   Download data T1Brain.mat (courtesy of brain web `http://mouldy.bic.mni.mcgill.ca/brainweb/` and do the following using Matlab. The variable $I$ (a $217 \times 181$ matrix) in T1Brain.mat is the image of one slice of a simulate T1 weighted human brain image. Write a report with all results listed and code attached.

   (a) Use $K = fftshift(fft2(I))$to simulate the corresponding k-space data (imagined to be scanned by MRI machine). Use

   ```
   figure;imagesc(log(abs(K)));axis image;colormap(gray)
   ```

   to visualize the k-space data. Also visualize the corresponding image $I$.

   (b) **Low pass filtering:** Define a square centered at the center of $K$ with side 61, define a new k-space data $K_l$ that keeps the frequency of $K$ at locations inside the square, and has zero values at other locations. Visualize both the k-space and corresponding image $I_l$ (use $abs(I_l)$ when it is complex-valued). How is this image different from that in (a)?

   (c) **High pass filtering:** Define a new k-space data $K_h$ that keeps all the frequency of $K$ at locations outside the square, and has zero values elsewhere. Visualize both the k-space data and corresponding image and summarize the difference with (a).

   (d) **Band pass filtering:** Define a new k-space data $K_b$ that keeps all the frequency of $K$ at location between the two squares centered at the center of the image and with sides 11 and 121. Do the above comparison.

   (e) **Half (partial) Fourier imaging:** The image $I$ is real-valued, explain how to reduce the scan time by a factor of 2 through only sampling half of the k-space data. Prove your approach works theoretically and numerically.

   (f) **Equidistance under-sampling:** Another way to speed up the scan is to scan every other row in k-space. Let $K_e$ be a new k-space data that keeps all odd rows of $K$ while zero others. Visualize both the new k-space data and image.

3. **Application of discrete Fourier transform in image compression.**

   (a) Find a color image online or from your digital camera and load it into Matlab with "$z$ = imread('myimage.jpg')". "imread" can also read files of other formats such as bmp,png,tif. $z$ has three channels. Visualize the color image using "figure;image($z$);axis image". Construct a grayscale image for image compression by combining the three channels : $zg = 0.2989 * double(z(:,:,1)) + 0.5870 * double(z(:,:,2)) + 0.1140 * double(z(:,:,3))$. Try visualize the grayscale image using "figure;image($z$);axis image" and see if you are happy with the visualization. Try the same thing again but with the right colormap: $L = 255; colormap([(0:L)/L; (0:L)/L; (0:L)/L;]')$. Learn how "colormap" works by "doc colormap" in Matlab. Another simpler way to visualize the grayscale image is "figure;imagesc(z);axis image;colormap(gray)". "axis image" makes sure the image is displayed in the original horizontal and vertical dimension ratio.

   (b) Compute the DFT of $zg$ with $Z = fft2(zg)$. Since DFT components might vary over many orders of magnitude, you can view its log function: $Zlog = log(1 + abs(Z))$; imagesc($Zlog$). In order to properly scale the image, it's helpful to compute the maximum value of $zlog$ with $M = max(max(Zlog))$; and then visualize "image($255 * Zlog/M$)". You can also visualize the fftshifted version by letting $Z = fftshift(fft2(zg))$. You should be able to see certain symmetries.

   (c) Choose a $thresh$ between 0 and 1. Let's start with $thresh = 0.0001$, we zero out all frequencies in $Z$ that fall below a value $thresh * M$ in magnitude. This is done with $Zthresh = (abs(Z) > thresh * M). * Z$; The compression ratio is defined as the fraction of the Fourier coefficients survived the cut, i.e. $sum(sum(abs(Zthresh) > 0))/m/n$ where $m, n$ are the dimension of the grayscale image $zg$. To uncompress and view the image, inverse transform with $zthresh = real(iffts(Zthresh))$; and view with "image($zthresh$)". The distortion of the compressed signal is: $100 * norm(zg - zthresh,'fro')^2/norm(zg,'fro')^2$; Repeat the computation above for threshold values $0.001, 0.01, 0.05$, or even larger values. In each case compute the compression ratio, the distortion, display the image, and rate its quality. Construct a plot of the distortion (vertical axis) versus compression ratio. It may be helpful to use a logarithmic scale on one or both axes.

4. **One-dimensional DFT.** Use the matrix $F_4^*$ to hand compute the inverse DFT of the vector $X = (3, 1+i, 1, 1-i)^T$.

5. Let $\mathbf{x}, \mathbf{y}$ be two column vectors of length $m$ and $n$ respectively, with one dimensional DFT $\mathbf{X}, \mathbf{Y}$. Let $\mathbf{z} = \mathbf{x}\mathbf{y}^T$ be the outer product of $\mathbf{x}$ and $\mathbf{y}$. Show that $\mathbf{Z} = \mathbf{X}\mathbf{Y}^T$.

6. **Two-dimensional DFT.** Let $\mathbf{e_{j,k}}$ denote an $m \times n$ matrix that has a 1 in the row $j$, column $k$ position and zeros elsewhere. Find the two-dimensional DFT of $\mathbf{e_{j,k}}$ and also the magnitude of each $DFT$ coefficient. (Hint: use the result of problem 4.)