ECE 5273
Homework 4

Spring 2022 Dr. Havlicek

The images in problems 1-5 are all of size 8×8. Unlike the previous assignments, the pixels in these images are floating point. In some cases, they are also complex-valued. The next two paragraphs provide some notes about handling these images in Matlab and in C.

Matlab: Since the default data type in Matlab is "complex double precision," you don't need any special data type declarations to handle these images. However, it is important for you to remember that the mathematics of the DFT requires zero-based row and column indexing. In other words, for the math to work it's important that the first row of the image is row ZERO and the first column is column ZERO. This is different from the Matlab convention for indexing arrays, where the first row has index ONE and the first column has index ONE. The Matlab statement

```
[COLS,ROWS] = meshgrid(0:7,0:7);
```

will make an 8×8 image (matrix) `COLS` where each pixel is equal to the zero-based column number and an 8×8 image `ROWS` where each pixel is equal to the zero-based row number. Then you can easily make the image $\mathbf{I}_1$ needed for problem 1 like this:

```
I1 = 0.5 * exp(j*2*pi/8*(2.0*COLS + 2.0*ROWS));
```

C: you should use float arrays to handle these images in C. For the complex-valued images, you should use one float array for the real part and another float array for the imaginary part. Alternatively, you could define a complex data type – but that takes more work.

1. Let $\mathbf{I}_1$ be defined by

$$I_1(m,n) = \frac{1}{2} \exp\left[j\frac{2\pi}{8}(u_0 m + v_0 n)\right],$$

where $m =$ **COLUMN** and $n =$ **ROW**. The horizontal frequency is $u_0 = 2$ cycles per image (cpi) and the vertical frequency is $v_0 = 2$ cpi. Show the real and imaginary parts of $\mathbf{I}_1$ as gray scale images with 8 bits per pixel (bpp) and full-scale contrast. Compute the DFT $\widetilde{\mathbf{I}}_1$ and center it so that the frequency origin is in the middle (this is done automatically by the C function `fft2d` available in the file `DoFFT.c` on the course web site under the "Code" link). Print out the real and imaginary parts of $\widetilde{\mathbf{I}}_1$ as 8×8 ascii floating point arrays. Whether you use Matlab, C, or some other language, be sure to format the output so that it is easy to read (the columns should line up and the rows should not "line-wrap"). In C, you should use the `printf` or `fprintf` statements to do this. In Matlab, if the 2D DFT is in the array `Itilde1`, then you can use the following statements:

```
fprintf(1,'%s\n','Re[DFT(I1)]:');
disp(round(real(Itilde1) * 10^4)*10^(-4));
fprintf(1,'%s\n','Im[DFT(I1)]:');
disp(round(imag(Itilde1) * 10^4)*10^(-4));
```

2. Let $\mathbf{I}_2$ be defined by

$$I_2(m,n) = \frac{1}{2} \exp\left[-j\frac{2\pi}{8}(u_0 m + v_0 n)\right],$$

where $u_0 = v_0 = 2$ cpi as in Problem 1. Show the real and imaginary parts of $\mathbf{I}_2$ as 8 bpp gray scale images with full-scale contrast. Compute the centered DFT $\widetilde{\mathbf{I}}_2$ and print out the real and imaginary parts of $\widetilde{\mathbf{I}}_2$ as 8×8 ascii floating point arrays.

3. Let $\mathbf{I}_3$ be defined by

$$I_3(m,n) = \cos\left[\frac{2\pi}{8}(u_0 m + v_0 n)\right],$$

where $u_0 = v_0 = 2$ cpi as in Problem 1. **Note:** $\mathbf{I}_3 = \mathbf{I}_1 + \mathbf{I}_2$. Show $\mathbf{I}_3$ as as an 8 bpp gray scale image with full-scale contrast (it should look the same as the real part of $\mathbf{I}_1$). Print out the real and imaginary parts of the centered DFT $\widetilde{\mathbf{I}}_3$ as 8×8 ascii floating point arrays.

4. Let $\mathbf{I}_4$ be defined by

$$I_4(m,n) = \sin\left[\frac{2\pi}{8}(u_0 m + v_0 n)\right],$$

where $u_0 = v_0 = 2$ cpi as in Problem 1. **Note:** $\mathbf{I}_4 = -j(\mathbf{I}_1 - \mathbf{I}_2)$. Show $\mathbf{I}_4$ as as an 8 bpp gray scale image with full-scale contrast (it should look the same as the imaginary part of $\mathbf{I}_1$). Print out the real and imaginary parts of the centered DFT $\widetilde{\mathbf{I}}_4$ as 8×8 ascii floating point arrays.

5. Let $\mathbf{I}_5$ be defined by

$$I_5(m,n) = \cos\left[\frac{2\pi}{8}(u_1 m + v_1 n)\right],$$

where $u_1 = 1.5$ cpi is the horizontal frequency and $v_1 = 1.5$ cpi is the vertical frequency. Show $\mathbf{I}_5$ as an 8 bpp gray scale image with full-scale contrast and print out the real and imaginary parts of $\widetilde{\mathbf{I}}_5$ as 8×8 ascii floating point arrays. Discuss the appearance of $\widetilde{\mathbf{I}}_5$ as compared to $\widetilde{\mathbf{I}}_1$ through $\widetilde{\mathbf{I}}_4$.

6. Obtain the images `camera.bin`, `salesman.bin`, `head.bin`, and `eyeR.bin` from the course web site. These are 256×256 grayscale images with 8 bpp and pixel values in the range 0 to 255. For each one, show the original image, the real part of the centered DFT,

the imaginary part of the centered DFT, the centered DFT log-magnitude spectrum, and the phase of the centered DFT as 8 bpp images with full-scale contrast.

7. Let $\mathbf{I}_6$ be the *Cameraman* image from the file `camera.bin`. Define a new image $\mathbf{J}_1$ according to $|\widetilde{\mathbf{J}}_1| = |\widetilde{\mathbf{I}}_6|$ and $\arg \widetilde{\mathbf{J}}_1 = 0$. Define another new image $\mathbf{J}_2$ according to $|\widetilde{\mathbf{J}}_2| = 1$ and $\arg \widetilde{\mathbf{J}}_2 = \arg \widetilde{\mathbf{I}}_6$. To within floating point roundoff, $\mathbf{J}_1$ and $\mathbf{J}_2$ will both be real (discard any nonzero imaginary part). $\mathbf{J}_1$ illustrates the contribution of the DFT magnitude to the original image, while $\mathbf{J}_2$ illustrates the contribution of the DFT phase to the original image.

   Show $\mathbf{J}_2$ as an 8 bpp grayscale image with full-scale contrast.

   $\mathbf{J}_1$ is a little bit trickier to display because there are a few pixels that are *much* brighter than the rest. But the pixels of $\mathbf{J}_1$ are all real-valued and greater than zero. So let $\mathbf{J}_1' = \log(\mathbf{J}_1)$ and display $\mathbf{J}_1'$ as an 8 bpp grayscale image with full-scale contrast.

DUE: 3/10/2022