BENG280A, Principles of Biomedical Imaging 10/02/12
Fall Quarter 2012

HOMEWORK #1
Due at 5 pm on Wednesday 10/10/12

Homework policy: Homeworks can be turned in during class or to the TA's mailbox in the Graduate Student Lounge. Late homeworks will be marked down by 20% per day. If you know that you need to turn in a homework late because of an emergency or academic travel, please let the TA know ahead of time. Collaboration is encouraged on homework assignments, however, the homework that you submit should reflect your own understanding of the material.

Readings: MRI notes by Lars Hanson (PDF on the website)

Problem 1: Consider the basis functions discussed in Lecture 2:
[1 1 1 1], [1 0 -1 0], [0 1 0 -1], and [1 -1 1 -1].

a) Professor X claims that these basis functions can be written as simple trigonometric functions defined on the domain x = {0, 1, 2, 3}. What are these trigonometric functions and what are their corresponding spatial frequencies?
b) Expand the vector [7 5 3 5] as the sum of these basis functions.
c) Expand the vector [7 0 3 0] as the sum of these basis functions.

Problem 2: Using the definition of the Fourier transform, derive the **scaling** theorem presented in class. Make sure to justify the use of the absolute value operation.

MATLAB exercises are on the next pages.

Matlab Exercise 1: In this exercise you will use MATLAB to plot out the complex function $g(x) = \exp(j2\pi k_x x)$.

Enter in the following MATLAB commands (you will want to save this in a script, so you can easily modify):

```
j = sqrt(-1);
x = -10:.1:10;
kx = .1;
g = exp(j*kx*2*pi*x);
figure(1);clf;
quiver3(x,zeros(size(x)),zeros(size(x)),zeros(size(x)),real(g),imag(g));
```

1) Describe the main features of the plot and explain the main features – i.e. what is the period of repetition and the "handedness of the helix". You will find it useful to use the "rotate" button on the MATLAB plotting interface to look at the helix from different views.
2) Which way do the arrows point at x = 0, x = 2.5 and x = 5.0? Explain how the directions match up with the analytical expressions of the function.
3) What happens when you make `kx = -0.1`? Explain what is going on.
4) What happens when you make `kx = 0.2`? Explain what you observe and compare this to the original plot (e.g. from part 1).

Next enter in the following commands:

```
span = -10:.5:10;
[x,y] = meshgrid(span,span);
kx = .1;ky = .1;
g = exp(j*2*pi*(kx*x + ky*y));
figure(2);clf;
quiver(x,y,real(g),imag(g));
grid;
```

1) Explain what you are seeing. What function are we plotting? How does the way in which we are plotting the function differ from the way we did it in the first half of this exercise?
2) What is the fundamental period of the object – does this match the analytical prediction?
3) Which way do the arrows point at (0,0), (5,5), and (2.5, 2.5)? Do these match the analytical prediction?
4) How does the plot change when you make kx = -0.1? Explain the differences.
5) How does the plot change when you make kx = 0.05? Explain the differences.
6) Try your own combination of values of kx and ky (different from what was done above) and explain the main features of what you are seeing.

Matlab Exercise 2: The purpose of this exercise is to familiarize you with the MATLAB functions used for performing 2D Fourier transforms and manipulating and displaying images. **Boldfaced** items should be turned in with the homework. You can always get more information about a command by typing *help <name of command>*, e.g., *help fft2*.

Steps:

1. First download the file BE280Ahw1im.mat from the course website.
2. Load the image into MATLAB with the command: *load BENG280Ahw1im*.
3. Type *whos* to see the variables in your MATLAB workspace. You should see a variable named *Mimage*. Type *size(Mimage)* to see how large the image is.
4. Use the command *imagesc(Mimage)* to display the image – you should see a sagittal image of a head. To change the colormap display to gray-scale, type *colormap(gray(256))* which will result in a display with 256 shades of gray. **Print out a copy of the image.** Try experimenting with different numbers for the colormap, e.g. *colormap(gray(20));*
5. Compute the 2D Fourier transform of the image with the command *Mf = fft2(Mimage);* where the 2D transform will now be stored in the variable *Mf*. Remember to add the semi-colon at the end of the command, otherwise MATLAB will display all the numbers in the matrix! The command *fft2* puts the zero-frequency value of the transform at the first indices of the matrix. For display it's convenient to put the zero-frequency value in the center of the matrix. To do this, type *Mf = fftshift(Mf);*
6. Type *imagesc(abs(Mf))* to display the magnitude of the transform. It will be hard to see anything, because the dynamic range of the Fourier transform is so large. To get a sense of the dynamic range, find the minimum value of the Fourier Transform with the command *min(min(abs(Mf)))* and the maximum value with the command *max(max(abs(Mf)))*. **Record these minimum and maximum values. What is the ratio of the maximum to minimum value?**
7. To scale the image so that you can get a better sense of what the Fourier transform looks like, you can use the *imagesc* command with the syntax: *imagesc(abs(Mf),[cmin cmax])* where *cmin* will be displayed as the darkest value on the image and *cmax* will be displayed as the lightest value on the image. For example, typing in *imagesc(abs(Mf),[200 1e6])* should give you a nice looking result. Experiment with different values of *cmin* and *cmax*.
8. You can also look at the real part, the imaginary part and the phase of the transform with the commands *imagesc(real(Mf))*, *imagesc(imag(Mf))*, and *imagesc(angle(Mf))*, respectively. If necessary you can use the *[cmin cmax]* option to scale the image properly. **Print out images of the magnitude and phase and real and imaginary parts of the transform. To plot more than one image on the same Figure, you can make use of the *subplot* command in MATLAB.**
9. Another way to look at the transform is to use the *mesh* command. Try *mesh(abs(Mf))*. It will be a little hard to see what is going on, so do the following: Define *span = 128 + (-20:20);* then type *mesh(abs(Mf(span,span)));*
10. ***Resolution***. What happens when we zero out the outer regions of the Fourier transform?
    (a) *Resolution reduction in the x-direction.*
    ```
    >> res_span = 129+(-16:16);
    >> Mf2 = zeros(256,256);
    >> Mf2(:,res_span) = Mf(:,res_span);
    >> Mf2 = fftshift(Mf2);
    >> M_resx = ifft2(Mf2);
    >> imagesc(abs(M_resx)); % This will show reduction of resolution in the x-direction.
    ```
    (b) Demonstrate resolution reduction in the y-direction**. Hand in code and image.**
    (c) Demonstrate resolution reduction in the x and y directions. **Hand in code and image**
11. ***Missing data in k-space***. We can also zero out the inner regions of the Fourier Transform.
    ```
    >>Mfzero = Mf;
    >>Mfzero(ky,kx) = 0;
    >>iMFzero= ifft2(fftshift(Mfzero)); % look at resulting image.
    ```
    Zero out the following:
    (a) kx = 129; ky = 129
    (b) kx= 1:256; ky = 129+(-16:16);
    (c) kx = 129+(-16:16); ky = 1:256;
    (d) kx = 129+(-16:16); ky = 129 + (-16:16);
    (e) kx = 1:2:256;ky = 1:256;
    For each set of parameters, plot out the Fourier transform and the resulting image. Give a qualitative explanation of why the image looks the way it does.

12. ***Spikes in the data***. You can put a spike at location (kx,ky) in Fourier space with the following commands

```
>>Mfspike = Mf;
>>spike = 100e6;
>>Mfspike(ky,kx) = Mf(ky,kx) + spike; % add spike
>>iMFspike = ifft2(fftshift(Mfspike)); % look at resulting image.
```

Put spikes at:

(a) kx = 129; ky = 129 – note this point corresponds to the center of k-space (i.e. kx = 0, ky = 0 in terms of the coordinates we used in class).

(b) kx= 161; ky = 129

(c) kx = 161; ky = 161

For each spike location, plot out the Fourier transform and the resulting image. Give a qualitative explanation for why the image looks the way it does. For example, what accounts for the direction of the artifacts?