BENG280A, Principles of Biomedical Imaging 09/27/04
Fall Quarter 2005

HOMEWORK #1
Due by 4 p.m. Wednesday 10/5/04

Homework Policy: Turn in assignments to the TA. Late homeworks will be marked down 20% for every day that they are late. 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.

Lecture Slides: PDF files of the next day's lecture slides will be posted on the course website by 5 p.m. the night before. Please make a printout before coming to class.

Readings:
Prince and Link: Read Chapter 12 and Chapter 13 (Sections 13.1 through 13.3.1; just skim 13.2.2 for now).

Problems:

1. Problem 12.1. For the time that you calculate for this problem, what will the phases be at z = 0.25 cm, 0.5 cm and 0.75 cm?
2. Problem 12.2.
3. Show that $M_z(t) = M_0 + \left(M_z(0) - M_0\right)e^{-t/T_1}$ is a solution to the differential equation $dM_z/dt = \dfrac{M_0 - M_z}{T_1}$
4. Problem 12.12.
5. Problem 13.11. In addition, make a sketch of the phases of the spins at 2 and 4 ms. You may use MATLAB and may find it useful to use the MATLAB function *quiver*.

MATLAB Exercise: 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 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)));* **Print out the meshplot and explain what this series of commands is doing.**