Cristian Mojica

Prof. Sereno

PSY 769

19 November 2022

Homework 2

1: 1-D FT. Plot the following 1-D functions and their Fourier amplitude spectra:

f1(x) = 0.3*cos(3*2π*x), f2(x) = 1.0*cos(8*2π*x), and f3(x) = f1(x) + f2(x).

Qualitatively describe the spectrum of the first two functions and then the spectrum of their sum.

```
step = 1/256; % [stuff after a percent sign is a comment]
x = 0:step:1-step; % get vector of 256 values of x from 0 to 1
```

```
f1 = 0.3 * cos(3*2*pi*x); % f1 is a real vector with, e.g., 24 cycles from 0-1
FT_f1 = fft(f1); % fft() returns the discrete fourier transform of f1
FT_f1s = fftshift(FT_f1); % fftshift displays zero freq of spect in middle of graph
plot(abs(FT_f1s)); % FT_f1 is complex vector; abs() gets element amplitudes
```

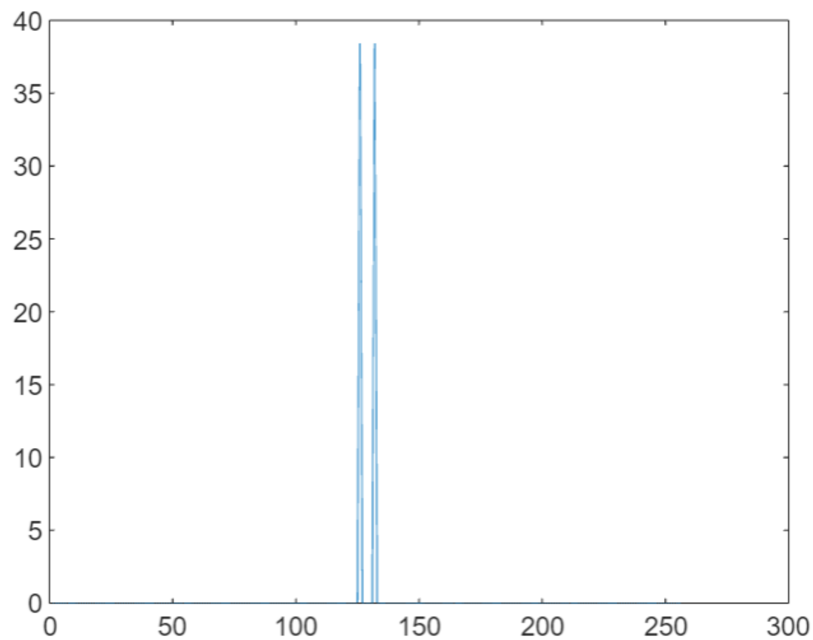

Figure 1: Fourier amplitude spectrum of f1

```
f2 = 1.0 * cos(8*2*pi*x); % f2 is a real vector with, e.g., 24 cycles from 0-1
FT_f2 = fft(f2); % fft() returns the discrete fourier transform of f2
FT_f2s = fftshift(FT_f2); % fftshift displays zero freq of spect in middle of graph
plot(abs(FT_f2s)); % FT_f2 is complex vector; abs() gets element amplitudes
```

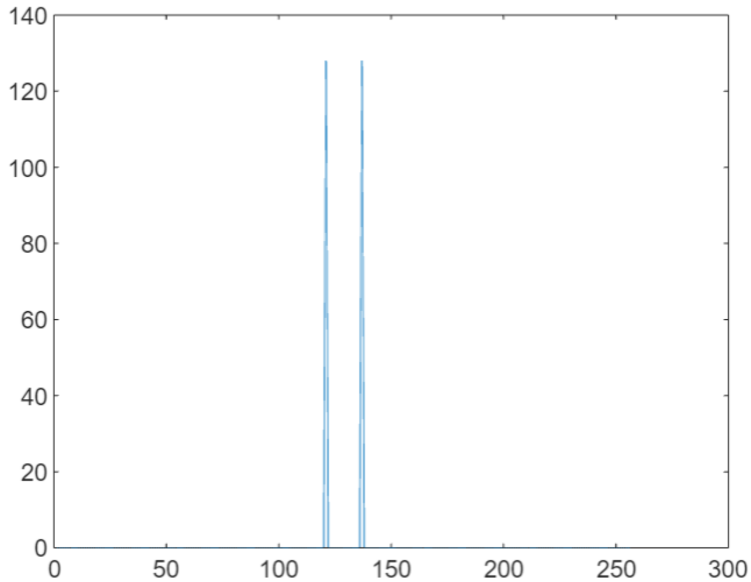

Figure 2: Fourier amplitude spectrum of f2

```
f3 = f1 + f2; % f3 is a real vector with, e.g., 24 cycles from 0-1
FT_f3 = fft(f3); % fft() returns the discrete fourier transform of f3
FT_f3s = fftshift(FT_f3); % fftshift displays zero freq of spect in middle of graph
plot(abs(FT_f3s)); % FT_f3 is complex vector; abs() gets element amplitudes
```

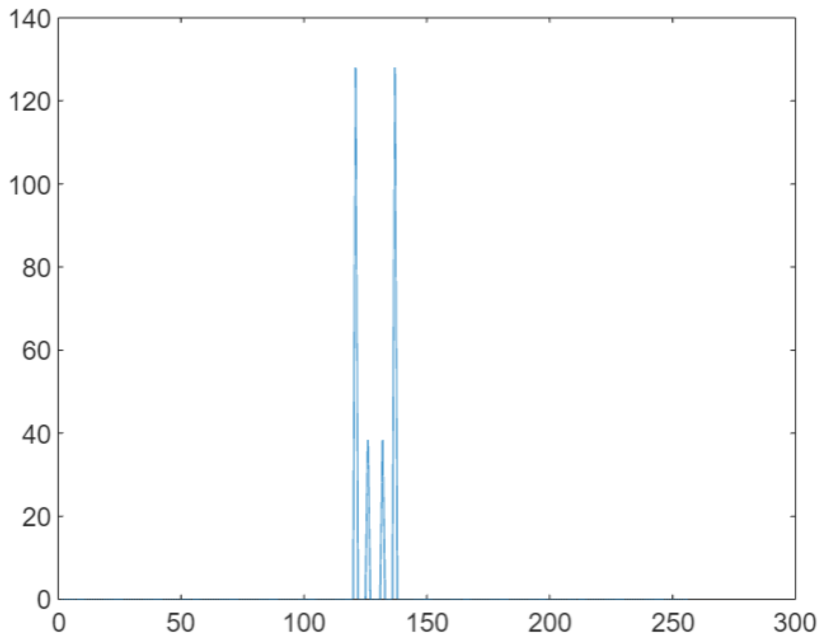

Figure 3: Fourier amplitude spectrum of f3

All three spectra consist of at least two decidedly centered and sharp spikes. In f1, these two spikes are close to each other, achieve a height below 40, and are between x = 100 and x = 150. In f2, the spikes are also between x= 100 and x = 150, but they are slightly further apart from each other and attain a height above 120. In f3, there are four spikes between 100 and 150. The same spikes from f1 and f2 and brought together exactly as they are into this one plot for f3.

2: 1-D Inverse FT. (this is a continuation of Problem 1). F1(f) is the frequency spectrum of f1(x). Reconstruct a function f'1(x) from F1(f) using the inverse Fourier transform, ifft(). Plot the real and imaginary parts, and the amplitude and phase of f'1(x). using real(), imag(), abs(), and angle(). Explain why the amplitude plot looks different than the real plot. The reconstructed phase and/or imaginary plots may be jagged; if so, explain why.

```
plot(real(ifft(FT_f1)));
```

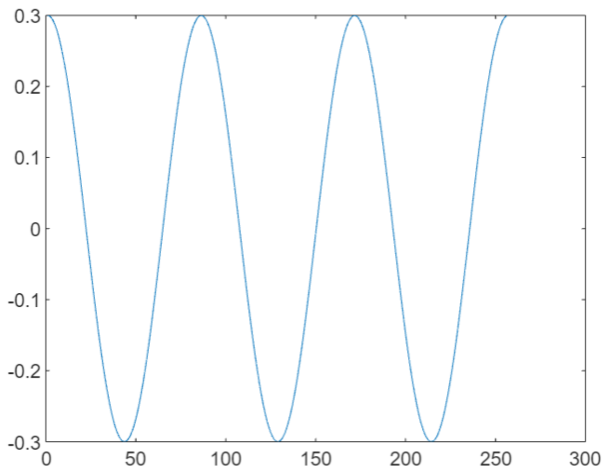

Figure 4: Real component of f'1(x)

```
plot(imag(ifft(FT_f1)));
```

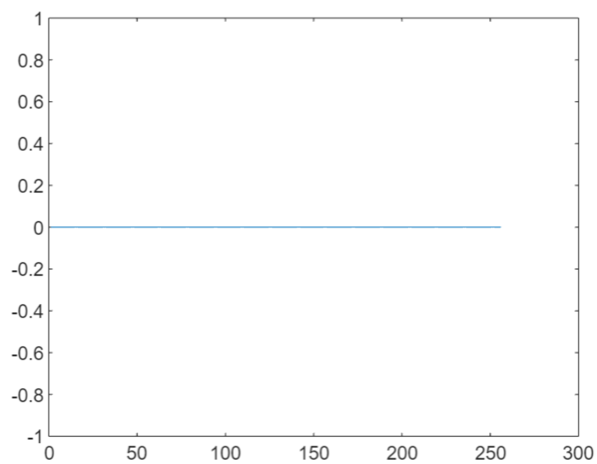

Figure 5: Imaginary component of f'1(x)

```
plot(abs(ifft(FT_f1)));
```

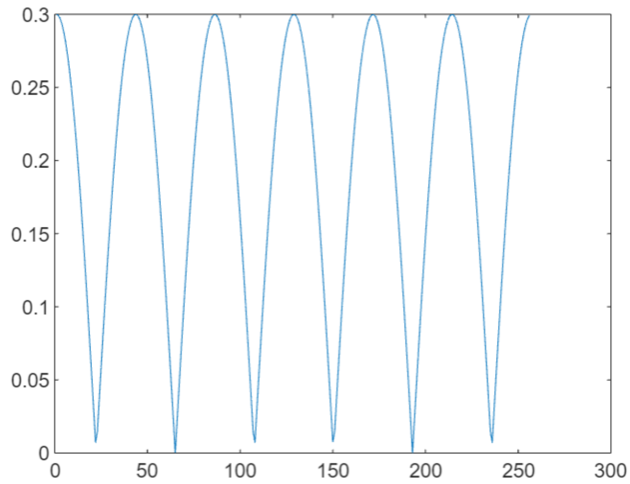

Figure 6: Amplitude component of f'1(x)

```
plot(angle(ifft(FT_f1)));
```

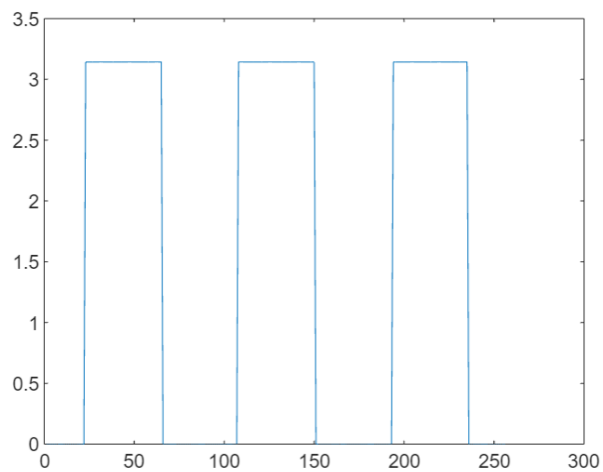

Figure 7: Phase component of f'1(x)

The amplitude plot looks different that the real plot because the amplitude plot displays the absolute values of the real plot (so the real plot was reflected across the x-axis whenever the function was negative in value to arrive at the amplitude plot). The phase plot shows three pulses of 180 degrees each, hence three distinct rectangles.

3: Display 2-D Image. Download the 256x256 sagittal T1 brain image tiff from http://www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff (or use any other, uncompressed, 256x256 grayscale brain slice tiff!) and convert it into a matrix with (MATLAB): Im=double(imread('t1sag.tiff')); Plot/display the image with: colormap gray; imagesc(Im,[minIm maxIm]); axis square; Play with different numbers for minIm and maxIm (don't forget the brackets, and use a space, not a comma to separate them). The function imagesc autoscales if you omit the [minIm maxIm] vector.

```
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
colormap gray, imagesc(Im,[0 100]), axis square;
```

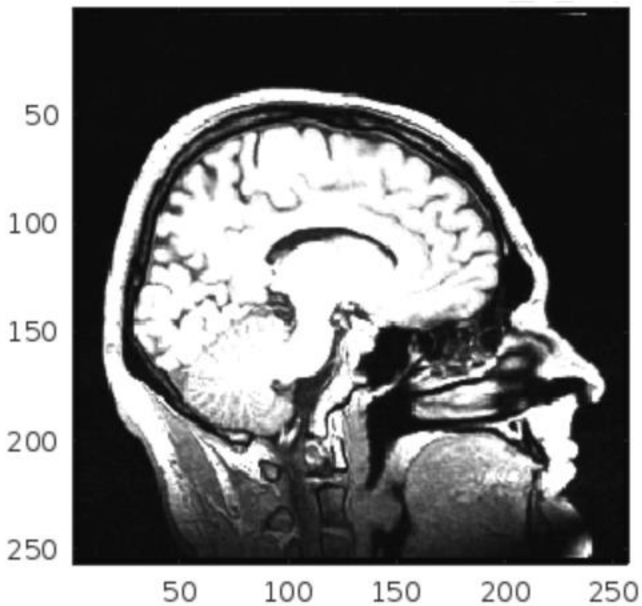

Figure 8: Contrast with Limits from 0 to 100

```
colormap gray, imagesc(Im,[0 200]), axis square;
```

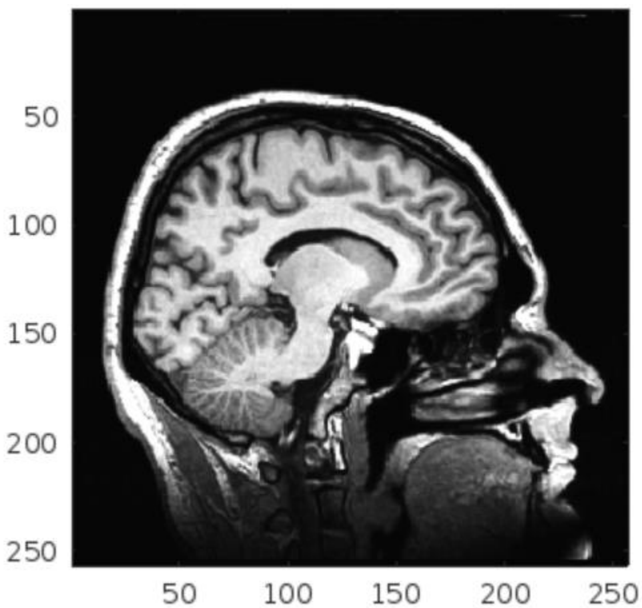

Figure 9: Contrast with Limits from 0 to 200

```
colormap gray, imagesc(Im,[70 100]), axis square;
```

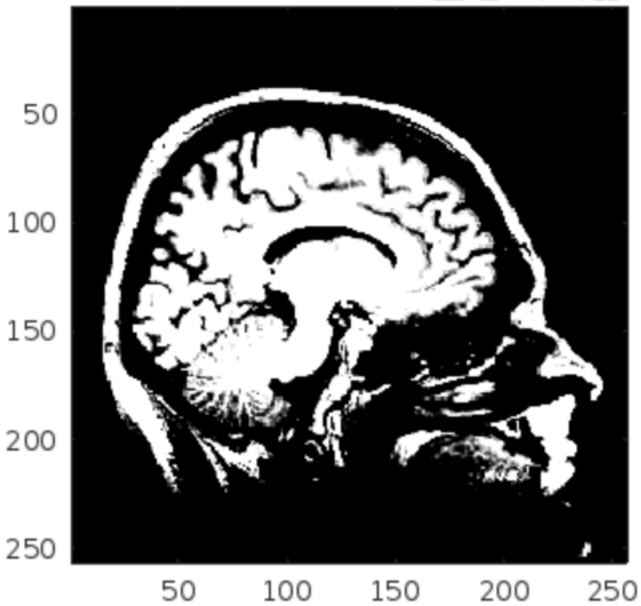

Figure 10: Contrast with Limits from 70 to 100

```
colormap gray, imagesc(Im,[70 200]), axis square;
```

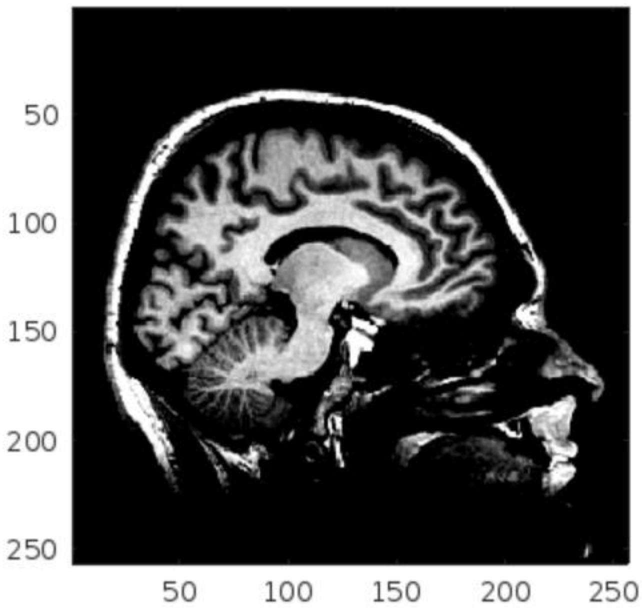

Figure 11: Contrast with Limits from 70 to 200

4: 2-D FT. Compute the 2-D Fourier transform of image Im using: FT=fftshift(fft2(Im)); The fftshift() function puts the zero spatial frequency in the middle for a matrix (2-D) as well as vector (1-D) (you can apply it to the image to see what it does). Then make four plots of FT (which is a 2-D matrix of complex numbers): first the real and the imaginary components, and then the corresponding amplitude and phase components. Use the functions: real(), imag(), abs(), angle() to extract the components and imagesc(Component,[minComponent maxComponent]); to plot them. You will have to experiment with the minimum and maximum values to make sense of the pictures. You can use the functions min() and max() (apply them twice, that is, recursively, to get one number out of a 2-D matrix!). Describe the resulting distributions in spatial frequency space (k-space).

```
figure; % multiple plots on one figure
Im = double(imread('t1sag.tiff'));
FT = fftshift(fft2(Im));
FT_Amp = abs(FT);
minAmp = min(min(FT_Amp)); maxAmp = max(max(FT_Amp));
colormap gray;
subplot(1,2,1); imagesc(Im); axis square;title('Image');
subplot(1,2,2); imagesc(FT_Amp,[minAmp 0.01*maxAmp]);axis square;title('K Ampl');
```

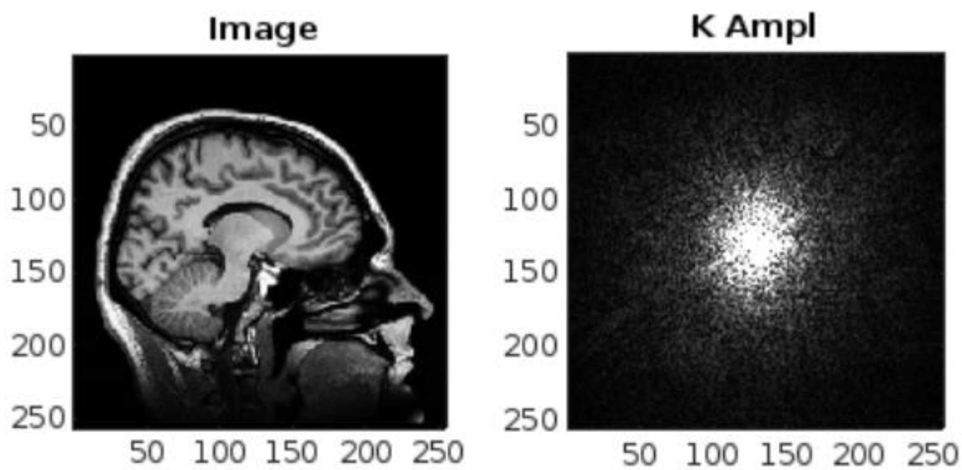

*Figure 12: Brain Image, witk K-Space Based on Limits minAmp to 0.01*maxAmp. Note the bright center of k-space, with the concentration of black spots increasing as distance from the center increases.*

```
figure; colormap gray;
subplot(1,2,1); imagesc(FT_Amp,[minAmp 0.01*maxAmp]);axis square;title('K Ampl -
FTAmp');
subplot(1,2,2); imagesc(FT_Amp);axis square;title('K Ampl - FTAmp');
```

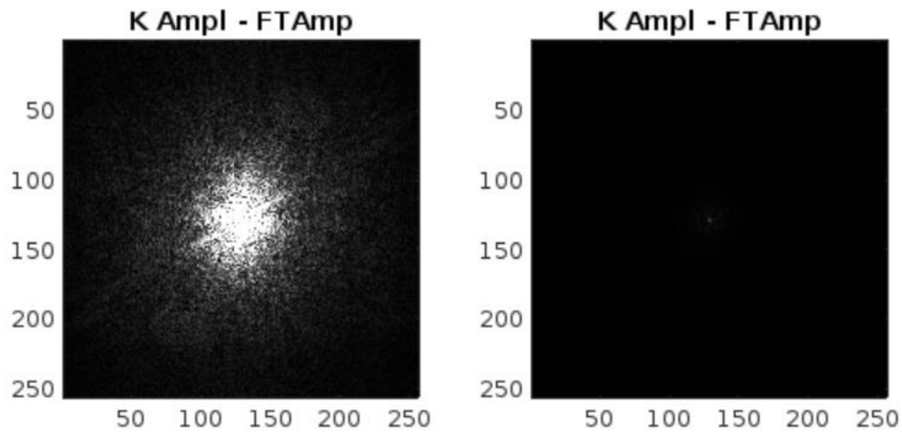

*Figure 13: Manipulating K-Space. left: minAmp to 0.01*maxAmp; right: autoscaled. The center of k-space gets darker with autoscaling, with primarily dark k-space in all other places. With scaling set from minAmp to 1/100 of maxAmp, the center of k-space is bright, with increasing concentration of black dots as the distance from the center increases.*

```
figure; colormap gray;
subplot(1,2,1); imagesc(real(FT),[minAmp 0.01*maxAmp]);axis square;title('K Ampl -
real');
subplot(1,2,2); imagesc(real(FT));axis square;title('K Ampl - real');
```

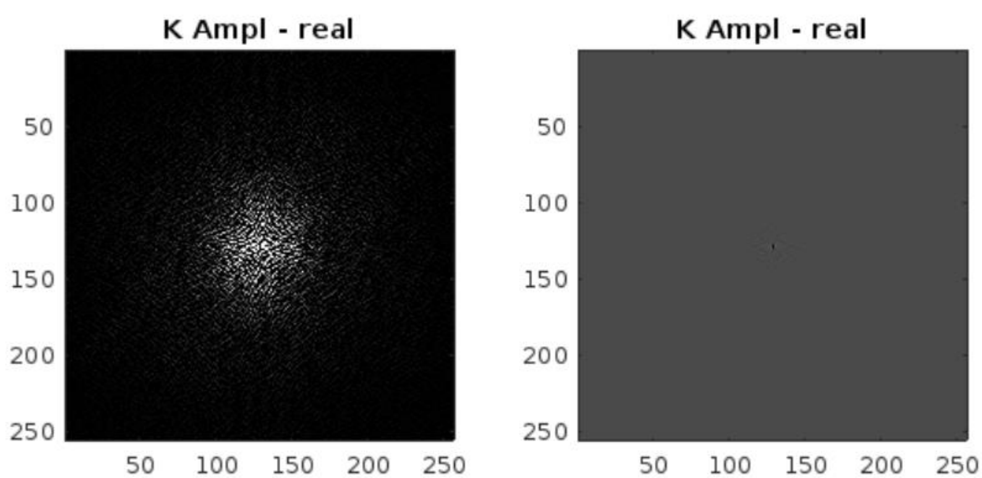

*Figure 14: Manipulating K-Space for the real component. left: minAmp to 0.01*maxAmp; right: autoscaled. Going from minAmp to 1/100 of maxAmp, the center of k-space is more obfuscated here since more black dots are present at the center, with the usual darkening present as the distance from the center increases. When autoscaled, k-space is primarily a dark gray, with what appears to be a single dark point at the center of k-space.*

```
figure; colormap gray;
subplot(1,2,1); imagesc(imag(FT),[minAmp 0.01*maxAmp]);axis square;title('K Ampl -
imag');
subplot(1,2,2); imagesc(imag(FT));axis square;title('K Ampl - imag');
```

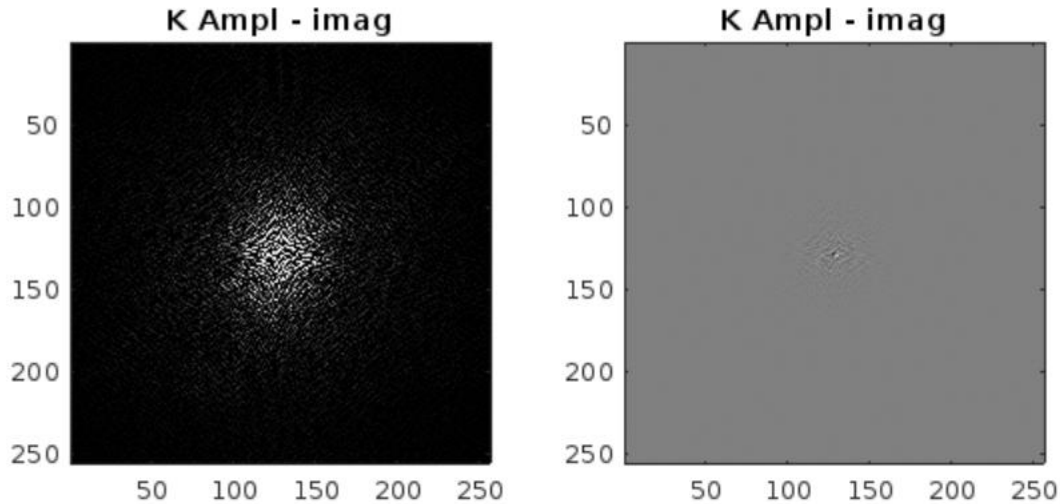

*Figure 15: Manipulating K-Space for the imaginary component. left: minAmp to 0.01*maxAmp; right: autoscaled. Going from minAmp to 1/100 of maxAmp, the center of k-space is more obfuscated here since more black dots are present at the center, with the usual darkening present as the distance from the center increases. When autoscaled, k-space is primarily a light gray, with what appears to be a single dark point and some rippling effect at the center of k-space.*

```
figure; colormap gray;
subplot(1,2,1); imagesc(abs(FT),[minAmp 0.01*maxAmp]);axis square;title('K Ampl -
abs');
subplot(1,2,2); imagesc(abs(FT));axis square;title('K Ampl - abs');
```

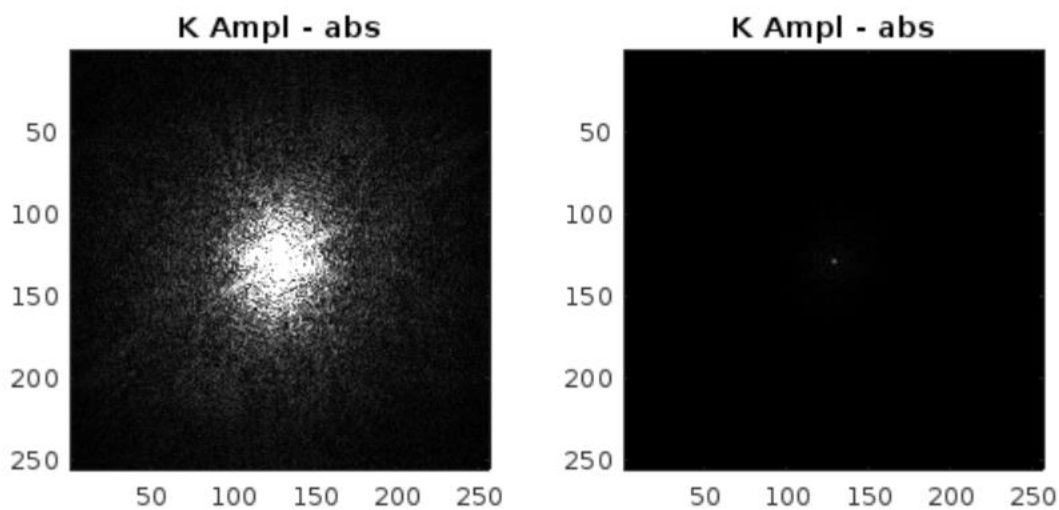

*Figure 16: Manipulating K-Space for the abs component. left: minAmp to 0.01*maxAmp; right: autoscaled. Going from minAmp to 1/100 of maxAmp, the center of k-space is bright, with the usual darkening present as the distance from the center increases. When autoscaled, k-space is primarily black, with what appears to be a single bright point at the center of k-space.*

```
figure; colormap gray;
subplot(1,2,1); imagesc(angle(FT),[minAmp 0.01*maxAmp]);axis square;title('K Ampl - angle');
subplot(1,2,2); imagesc(angle(FT));axis square;title('K Ampl - angle');
```

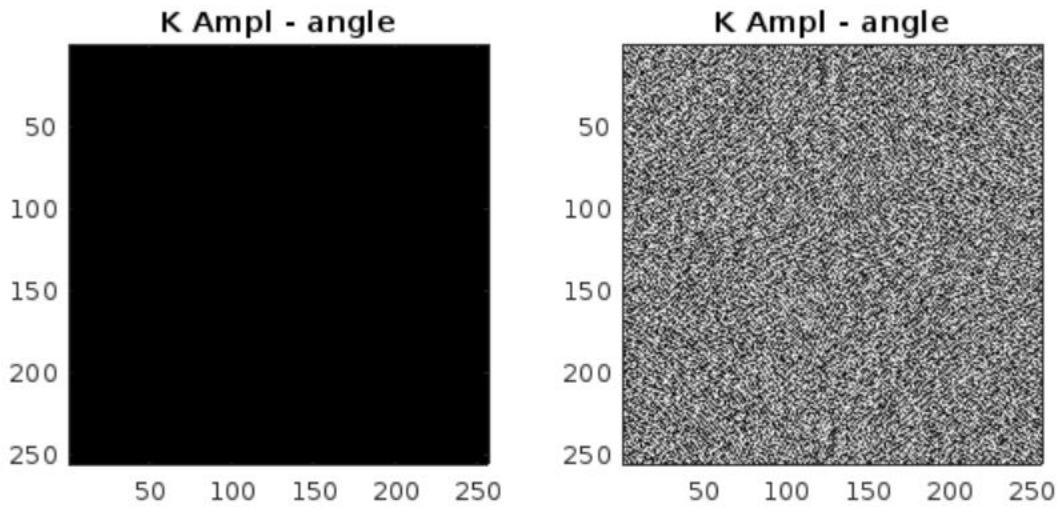

*Figure 17: Manipulating K-Space for the angle component. left: minAmp to 0.01*maxAmp; right: autoscaled. Going from minAmp to 1/100 of maxAmp, virtually all of k-space is black. When autoscaled, k-space is a homogeneous mixture of white, gray, and black (much like a static screen on a television set, but not with the definition or ridges of a fingerprint scan).*

```
colormap gray;
subplot(1,2,1); imagesc(Im); axis square;title('Image');
subplot(1,2,2); imagesc(FT_Amp,[0 500]);axis square;title('K Ampl');
```

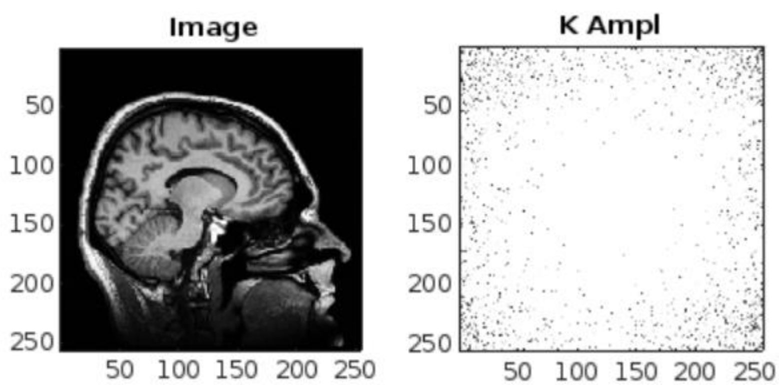

Figure 18: Manipulating K-Space for FT_Amp from 0 to 500. Note that k-space is primarily bright, with some dark spots present scattered around the corners and increasingly sparse as they approach the center of k-space.

```
colormap gray;
subplot(1,2,1); imagesc(Im); axis square;title('Image');
subplot(1,2,2); imagesc(FT_Amp,[300 500]);axis square;title('K Ampl');
```

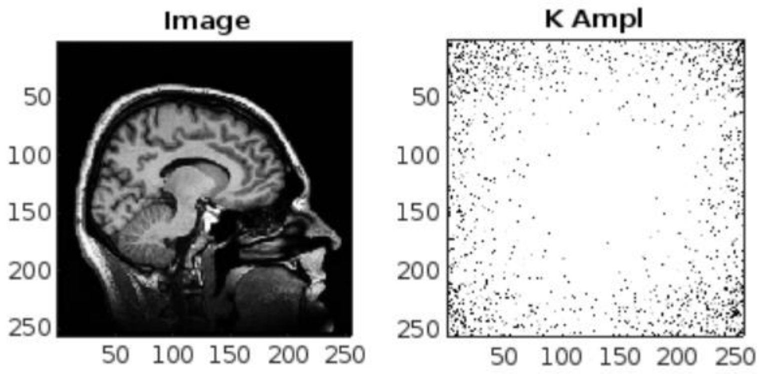

Figure 19: Manipulating K-Space for FT_Amp from 300 to 500. Note that k-space is primarily bright, with some dark spots (darker than in the scenario depicted in Figure 17) present scattered around the corners and increasingly sparse as they approach the center of k-space.

```
colormap gray;
subplot(1,2,1); imagesc(Im); axis square;title('Image');
subplot(1,2,2); imagesc(FT_Amp,[500 2000]);axis square;title('K Ampl');
```

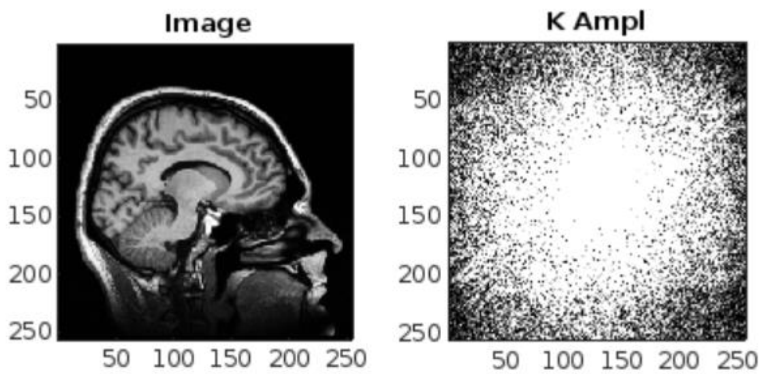

Figure 20: Manipulating K-Space for FT_Amp from 500 to 2000. Note that the center of k-space is still bright, but more black spots surrounding the center detract from its previous clear quality. There are more dark spots present as distance from the center increases, and dark spots appear with greater concentration in the corners.

5: k-Space Center. Manipulate the amplitude of the center point of k-space: F(kx0,ky0), which in the present case can be referenced with FT(129,129) (N.B.: those coordinates assume fftshift() has first been applied). First triple the center point, reconstruct the images by using ifft2(ifftshift()), plot the original and reconstructed amplitude image using the same maximums and minimums, and describe the result. Then do the same reconstruction after zeroing the center k-space point. Finally, multiply the original k-space center by -3. Briefly describe how the center of k-space affects contrast.

```
% Problem 5
figure; % multiple plots on one figure
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
minAmp = min(min(Im)); maxAmp = max(max(Im));
colormap gray;
FT = fftshift(fft2(Im));
FT(129,129) = FT(129,129)*3;
subplot(1,2,1);imagesc(Im, [minAmp maxAmp]); title("Original")
subplot(1,2,2);imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("K-Space
Center x 3")
```

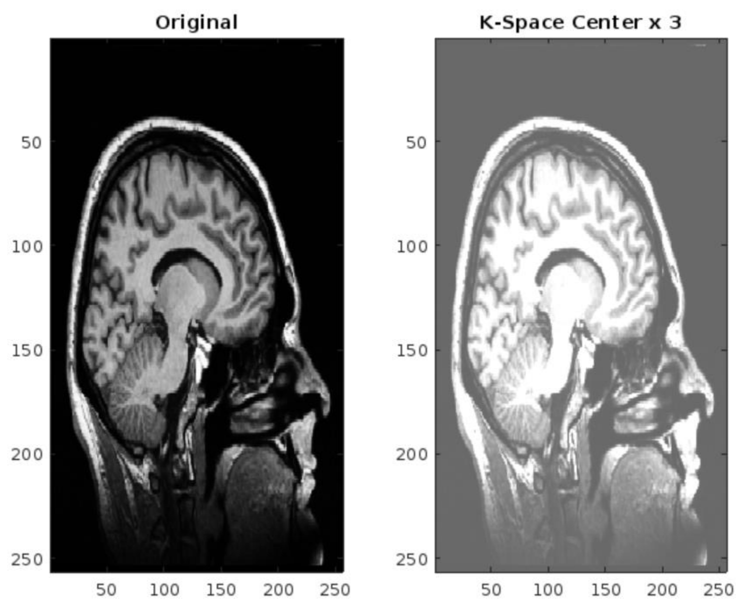

Figure 21: After multiplying the center of k-space by 3, the contrast of this image is generally lessened. Distinction of brain regions is still present, but to a slightly lesser extent than in the original.

```
figure; % multiple plots on one figure
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
minAmp = min(min(Im)); maxAmp = max(max(Im));
colormap gray;
FT = fftshift(fft2(Im));
FT(129,129) = FT(129,129)*0;
subplot(1,2,1);imagesc(Im, [minAmp maxAmp]); title("Original")
subplot(1,2,2);imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("K-Space
Center x 0")
```

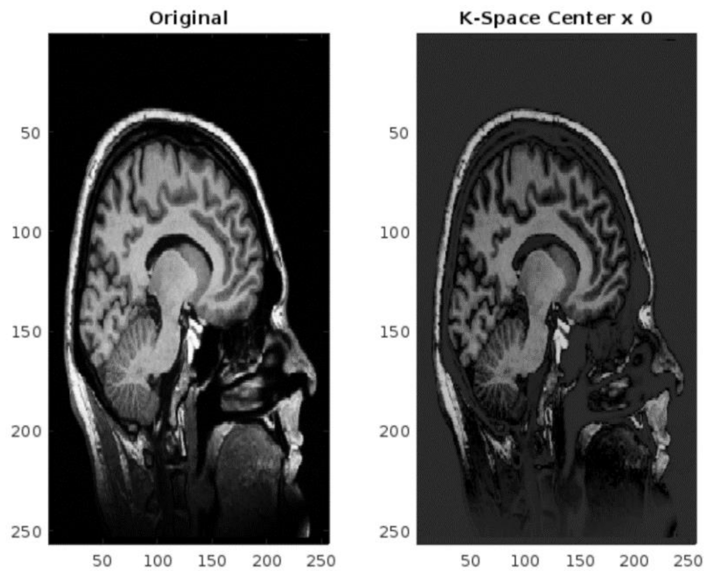

Figure 22: After multiplying the center of k-space by 0, the contrast of this image is generally greatened since the image is darker. Distinction of brain regions is still present. Some parts of what appears to be the nasal cavity are less distinct in the altered image.

```
figure; % multiple plots on one figure
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
minAmp = min(min(Im)); maxAmp = max(max(Im));
colormap gray;
FT = fftshift(fft2(Im));
FT(129,129) = FT(129,129)*-3;
subplot(1,2,1);imagesc(Im, [minAmp maxAmp]); title("Original")
subplot(1,2,2);imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("K-Space
Center x (-3)")
```

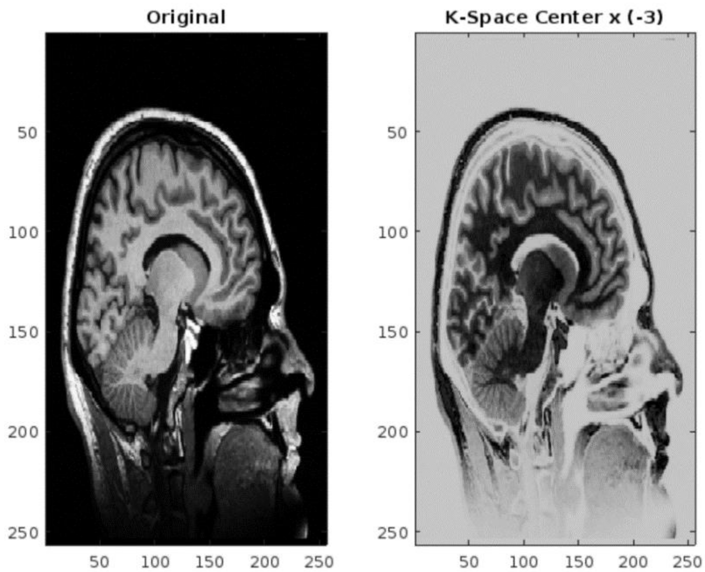

Figure 23: After multiplying the center of k-space by -3, the contrast of this image is completely inverted. The brain appears black in its neural matter and gray in its folds in the cortex, and air is captured in all gray compared to the usual all black.

6: Spikes in k-Space. An individual data point in k-space is sometimes mistakenly assigned a very large value (e.g., as a result of an electrical transient at the exact moment that the data point was being collected). Modify the following three k-space points by setting them to a large value (e.g., 107), one at a time:

(a) FT(129+5, 129) (ky=5, kx=0)

(b) FT(129, 129+33) (ky=0, kx=33)

(c) FT(129+5, 129+33) (ky=5, kx=33)

Reconstruct the images in each case using ifft2(ifftshift()) and plot and describe the effects of each manipulation.

```
figure;
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
FT = fftshift(fft2(Im));
FT_Amp = abs(FT);
minAmp = min(min(Im)); maxAmp = max(max(Im));
colormap gray;
FT(129+5, 129) = 10^7;
imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("FT(129+5, 129) = 10^7")
```

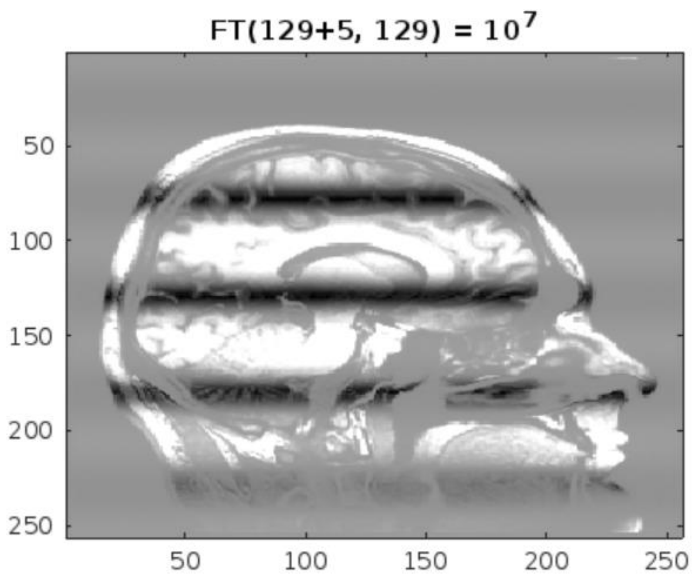

Figure 24: After setting (5,0) in K-Space equal to 10^7, the image's contrast is greatly dimmed, and five horizontal stripes appear across the image (four black, one gray running across the air represented by the gray background).

```
figure;
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
FT = fftshift(fft2(Im));
FT_Amp = abs(FT);
minAmp = min(min(Im)); maxAmp = max(max(Im));
colormap gray;
FT(129, 129+33) = 10^7;
imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("FT(129, 129+33) = 10^7")
```

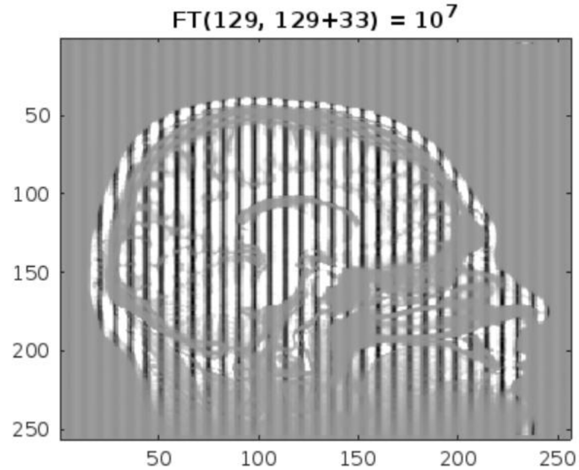

Figure 25: After setting (0,33) in K-Space equal to 10^7, the image's contrast is greatly dimmed, and 33 vertical stripes appear across the image (the stripes are black when crossing through the brain, but grey when crossing through air).

```
figure;
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
FT = fftshift(fft2(Im));
FT_Amp = abs(FT);
minAmp = min(min(Im)); maxAmp = max(max(Im));
colormap gray;
FT(129+5, 129+33) = 10^7;
imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("FT(129+5, 129+33) = 10^7")
```

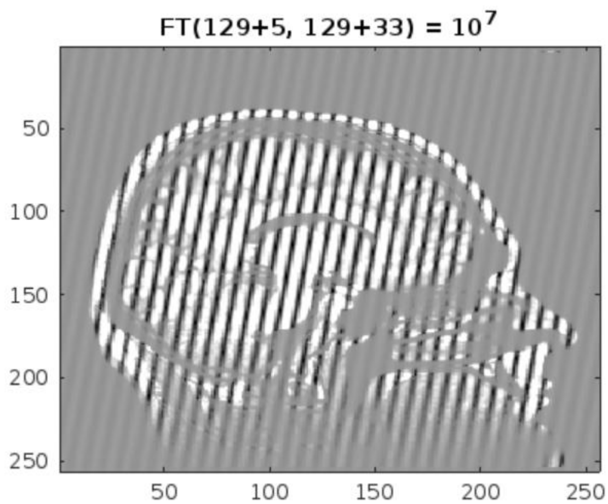

Figure 26: After setting (0,33) in K-Space equal to 10^7, the image's contrast is greatly dimmed, and 33 vertical stripes appear across the image (the stripes are black when crossing through the brain, but grey when crossing through air), but this time they cross through the brain at an angle.

7: Zero Portions of k-Space. By setting portions of k-space to zero, certain ranges of spatial frequency will be removed when the image is reconstructed. Set the following regions of k-space to zero, reconstruct the images as above, then plot k-space (amplitude) and the reconstructed image in each case, and comment on the result:

(a) set FT(ky, kx) = 0, where both kx and ky are between 129-32 and 129+32 (high pass)

(b) set FT(ky, kx) = 0, except where kx and ky are between 129-32 and 129+32 (low pass)

(c) set FT(ky, kx) = 0, when x is between 193 and 256 (zero right edge of k-space).

```
figure;
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
FT = fftshift(fft2(Im));
FT_Amp = abs(FT);
minAmp = min(min(Im)); maxAmp = max(max(Im));
colormap gray;
center = (129-32):(129+32);
 mask = zeros(256,256);
 mask(center,center) = 1;
 FT = FT .* mask;
imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("High pass")
```

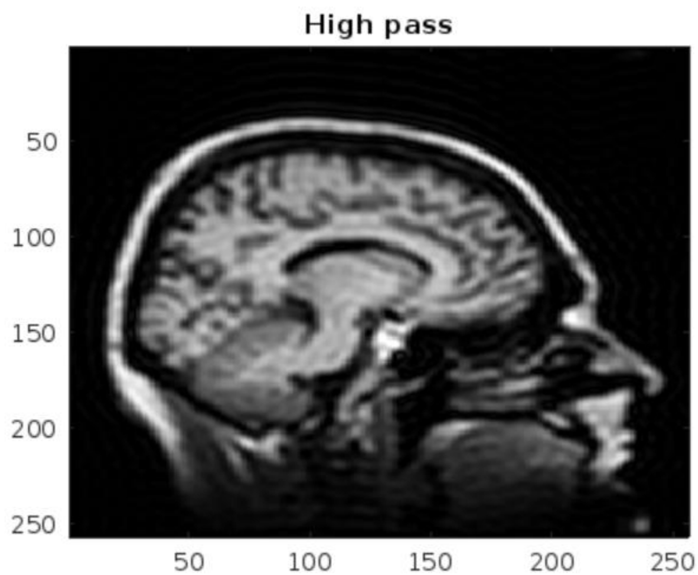

Figure 27: After applying a high pass filter to the original image, the image still has its usual contrast scheme, but the image is slightly blurry.

```
figure;
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
FT = fftshift(fft2(Im));
FT_Amp = abs(FT);
minAmp = min(min(Im)); maxAmp = max(max(Im));
colormap gray;
center = (129-32):(129+32);
 mask = ones(256,256);
 mask(center,center) = 0;
 FT = FT .* mask;
imagesc(abs(ifft2(ifftshift(FT)))); title("Low pass")
```

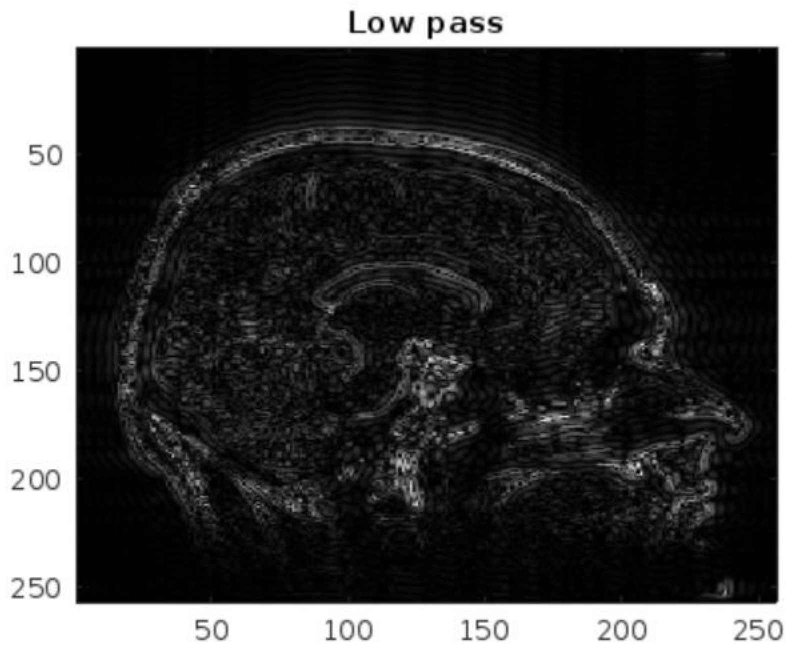

Figure 28: After applying a low pass filter to the original image, the image has been saturated in black points, with some gray present across the image. They imply some direction in the folding of cortex, but individual regions of cortex are harder to distinguish now.

```
figure;
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
FT = fftshift(fft2(Im));
FT_Amp = abs(FT);
minAmp = min(min(Im)); maxAmp = max(max(Im));
colormap gray;
center = (129-32):(129+32);
 mask = ones(256,256);
 mask(193:256, 1:256) = 0;
 FT = FT .* mask;
imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("Zero right edge of k-
space")
```

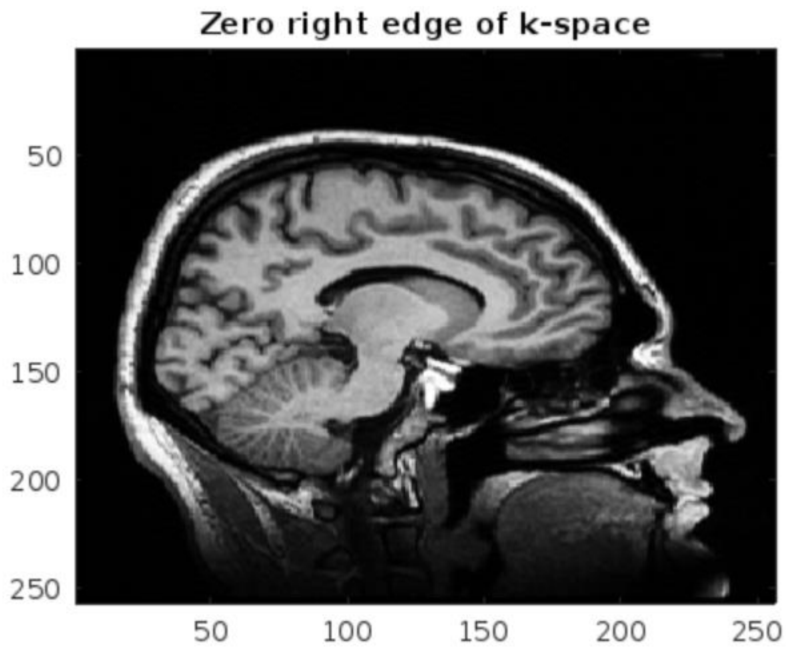

Figure 29: After zeroing out the underlying Fourier transform matrix where x is between 193 and 256, the resulting image is very similar to the original image, as if no alteration was made.

8: Subsample k-Space. If an image (or a time signal) is not sampled frequently enough, aliasing (wraparound) will occur in the frequency domain (that is, after a Fourier transform). This is also true when going from the frequency domain back to space (or time); that is, if k-space is not sampled frequently enough, aliasing will result in the image (or time) domain. Simulate this by zeroing every other line in k-space (e.g., even numbered lines). Comment on the effect of this undersampling after reconstructing the images with ifft2(fftshift()).

```
figure;
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
FT = fftshift(fft2(Im));
FT_Amp = abs(FT);
minAmp = min(min(Im)); maxAmp = max(max(Im));
ev = 2:2:maxAmp;
FT(ev, ev) = 0;
colormap gray;
imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("Even lines in k-space zeroed out")
```

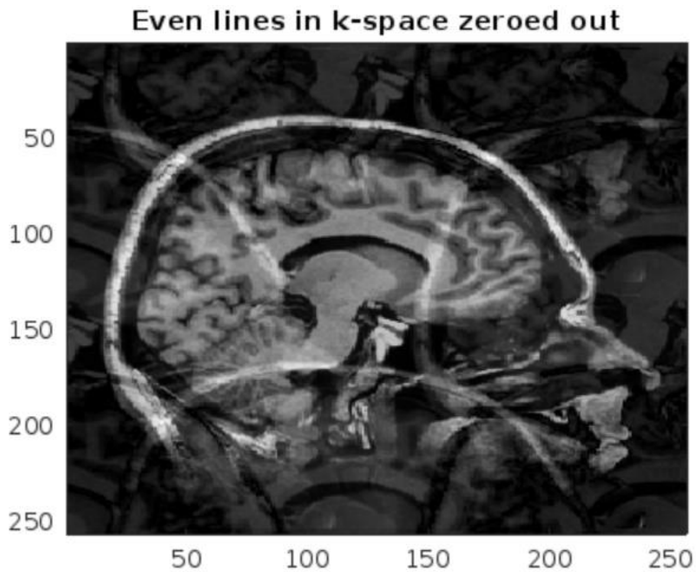

Figure 30: After zeroing out the even lines of k-space, there are faint, transparent wraparound ghosts of the brain image that appear around the main brain image.

```
figure;

Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
FT = fftshift(fft2(Im));
FT_Amp = abs(FT);
minAmp = min(min(Im)); maxAmp = max(max(Im));
od = 1:2:maxAmp;
FT(od, od) = 0;
colormap gray;
imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("Odd lines in k-space zeroed out")
```

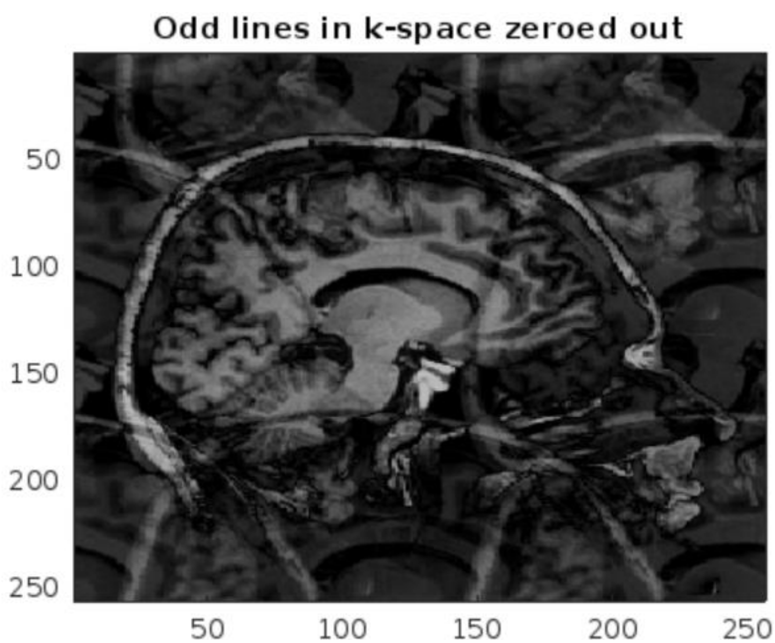

Figure 31: After zeroing out the odd lines of k-space, contrast is generally slightly darker than in the scenario presented in Figure 30, and there are faint wraparound ghosts of the brain image that appear around the main brain image.

9: Shift Alternate Lines of k-Space. When k-space data is collected during an EPI scan, the even and odd lines may not be properly aligned because of imperfections of the gradients. Simulate this by shifting even k-space lines to the left and the odd k-space lines to the right (do this for two different cases using the shifts given in (a) and (b)):

(a) set FT(kx, ky) = FT(kx-1, ky), when ky is odd and FT(kx+1, ky), when ky is even
(b) set FT(kx, ky) = FT(kx-4, ky), when ky is odd and FT(kx+4, ky), when ky is even

Plot both k-space and reconstructed images for the above manipulations. How do the wraparound ghosts subtly differ from the ones generated in the previous problem?

```
figure;
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
FT = fftshift(fft2(Im));
FT_Amp = abs(FT);
minAmp = min(min(Im)); maxAmp = max(max(Im));
FT(1:255,od) = FT(2:256, od);
FT(2:256,ev) = FT(1:255, ev);
colormap gray;
imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("Translation by One Unit")
```

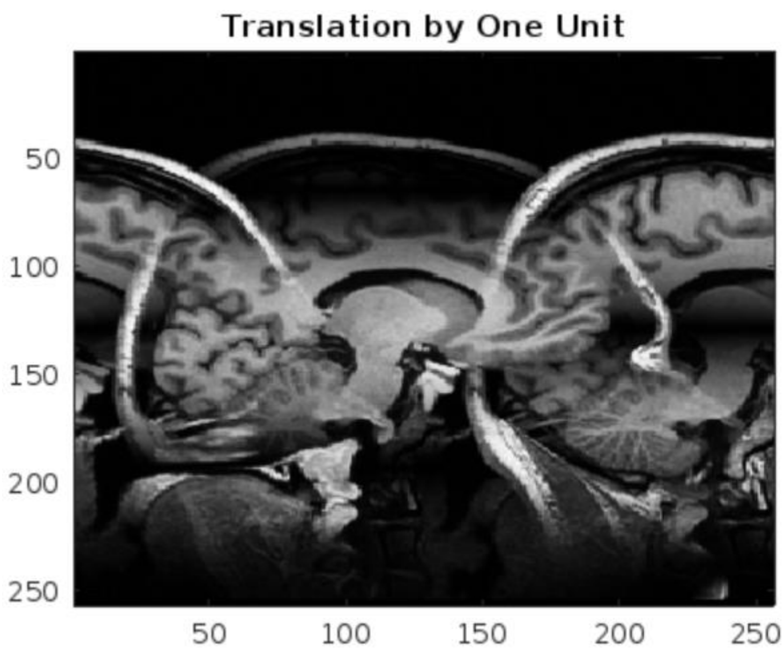

Figure 32: After a translation of K-Space lines by one unit, there are wraparound ghosts that appear, but they are not as faint as in Problem 8, and they only appear in a horizontal line instead of around the entire image. Also, some of the cortex at the top of the main brain image is obscured by darkness more than in Problem 8.

```
figure;
Im = double(imread('http:// www.cogsci.ucsd.edu/~sereno/276/t1sag.tiff'));
FT = fftshift(fft2(Im));
FT_Amp = abs(FT);
minAmp = min(min(Im)); maxAmp = max(max(Im));
FT(1:252,od) = FT(5:256, od);
FT(5:256,ev) = FT(1:252, ev);
colormap gray;
imagesc(abs(ifft2(ifftshift(FT))), [minAmp maxAmp]); title("Translation by Four Units")
```

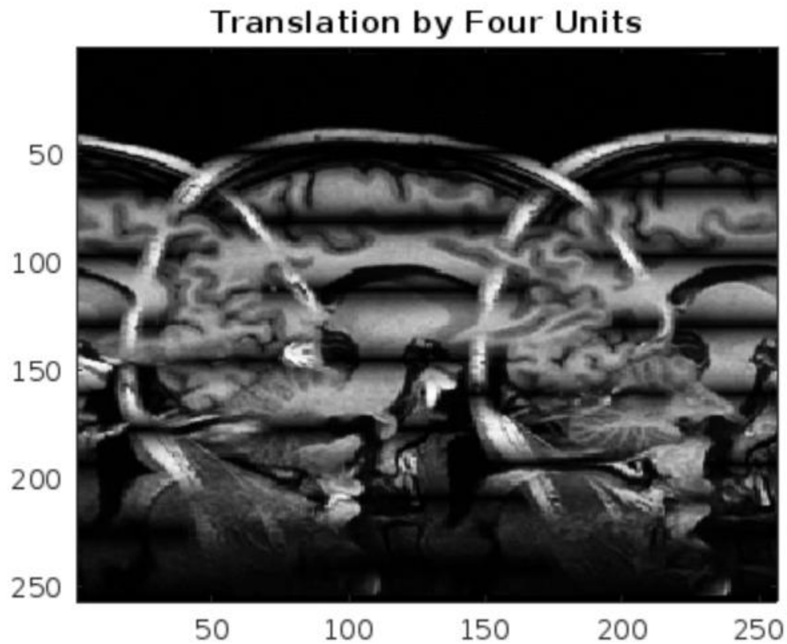

Figure 33: After a translation of K-Space lines by four units, there are wraparound ghosts that appear, but they are not as faint as in Problem 8, and they only appear in a horizontal line instead of around the entire image. Black lines horizontally cross through the main and wraparound images, and four such lines are present.