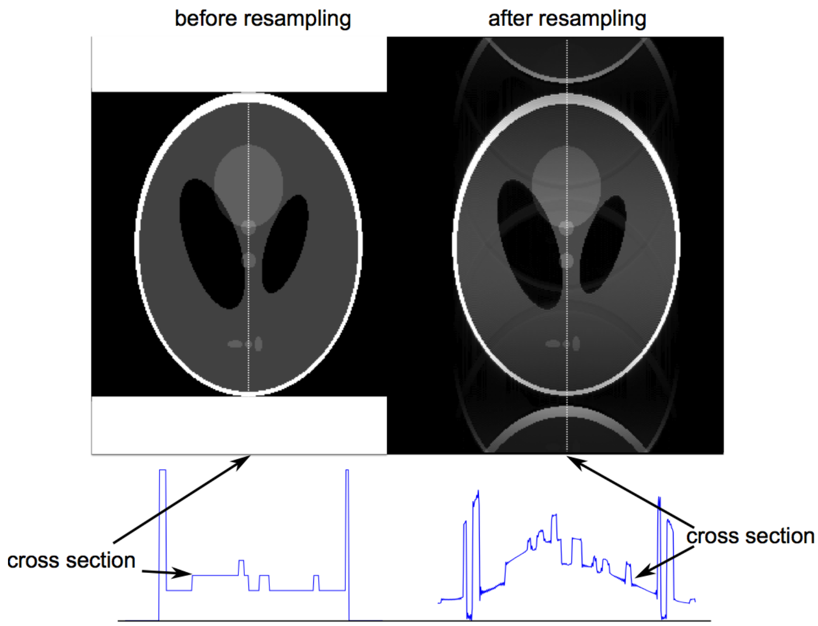
EE225E/BIOE265 Spring 2013 Miki Lustig
Principles of MRI

Assignment 2

Due Tuesday Feb 12, 2013

1. Read Nishimura Ch. 3

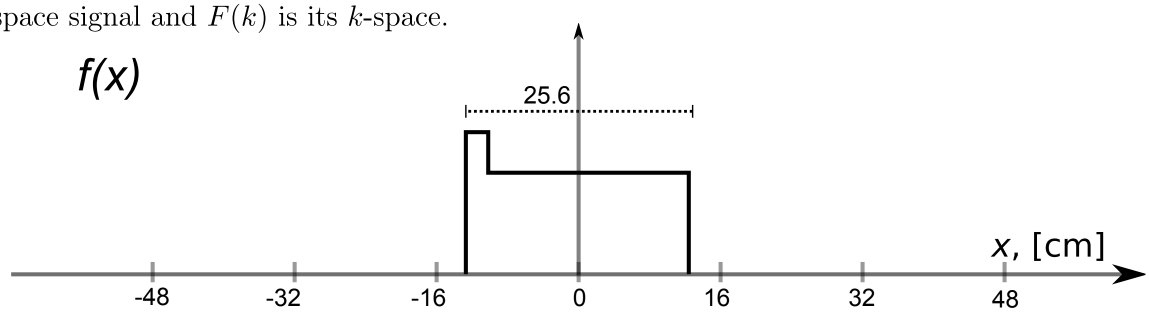
2. **Interpolation in k-space:** In this problem we will see the effect of linear interpolation in k-space. This is a more involved version of the question that was given in the midterm in 2010. The following image was resampled in the vertical direction in k-space by linear interpolation at $\frac{3}{4}$ the original rate.



Interpolation in k-space corresponds to finer k-space sampling and therefor a larger FOV. So, the FOV of the interpolated image (right) is larger in the vertical dimension.

There are two distinct artifacts that appear in the interpolated image and not in the original. The interpolated image is apodized (no longer has flat magnitude). In addition, there are some aliasing artifacts that appear as replication of the image in the interpolated dimension. These artifacts are the result of using a linear interpolation as opposed to a better interpolation scheme. In the following questions we will learn the source of these artifacts through a simple 1D example.

You are given the following 1D signal that you want to sample in k-space. In this question, $f(x)$ is the space signal and $F(k)$ is its k-space.



a) You decide to collect 256 k-space samples of $F(k)$ with $\Delta k = \frac{8}{256}$ cycles/cm. Express $\hat{f}(x)$, the result of sampling $F(k)$ in k-space, as a function of $f(x)$. Draw the signal $\hat{f}(x)$. Emphasize any artifacts in the drawing and point out the differences with the original $f(x)$.

b) You prepare to show the results from part (a) to your adviser, however, you suddenly remember that he asked you to sample k-space at a finer interval of $\Delta k = \frac{6}{256}$ cycles/cm. Not wanting to repeat the experiment, you decide to interpolate $\hat{F}(k)$ and resample it accordingly. Write an expression for $\tilde{F}(k)$, the resampled signal, assuming linear interpolation. Express $\tilde{F}(k)$ as a function of $\hat{F}(k)$,
HINT: linear interpolation is a convolution with an appropriately scaled triangle function $\Lambda(ak) = \begin{cases} 1 - |ak| & |ak| < 1 \\ 0 & \text{otherwise} \end{cases}$

c) Write an expression for $\tilde{f}(x)$, the image domain representation of $\tilde{F}(k)$ from part (b). First express $\tilde{f}(x)$ as a function of $\hat{f}(x)$ and then as a function of $f(x)$. Draw $\tilde{f}(x)$, pointing out any artifacts due to the interpolation and resampling.
(*HINT: First try to draw the effect of the interpolation, then draw the effect of the resampling.*)

d) Linear interpolation can be considered as interpolating with a poor low-pass filter. How would the result change if we use a filter with 0.1% ripple in the pass-band and stop band for the interpolation? Draw the result pointing out the differences.

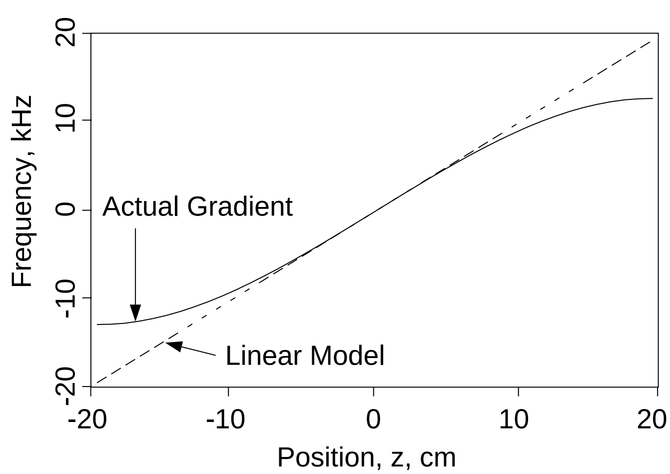
3. **RF Field**

   (a) Find the amplitude of an RF pulse that performs a 90 degree excitation in exactly 1 ms at 1.5T.

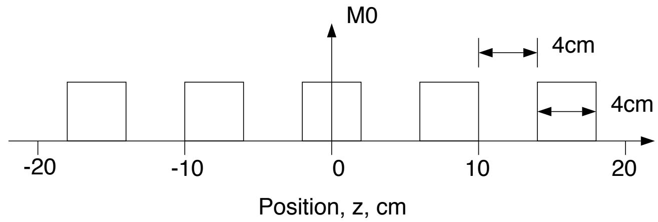
   (b) Find the amplitude of an RF pulse that performs a 90 degree excitation in exactly 1 ms at 3T.

4. **Non-Linear gradients.** One of the key elements in MRI is the use of a gradient field G which establishes a linear relationship between resonance frequency and position. While the linear model is convenient for analysis, real gradients are seldom exactly linear. In this problem we will look at some of the consequences of gradient non-linearity. This is a REAL situation in every scanner!

   (a) Consider a gradient system with the response shown by the solid line. A linear model is shown as the dashed line.



   Assume the object is a sequences of rectangles of uniform intensity.



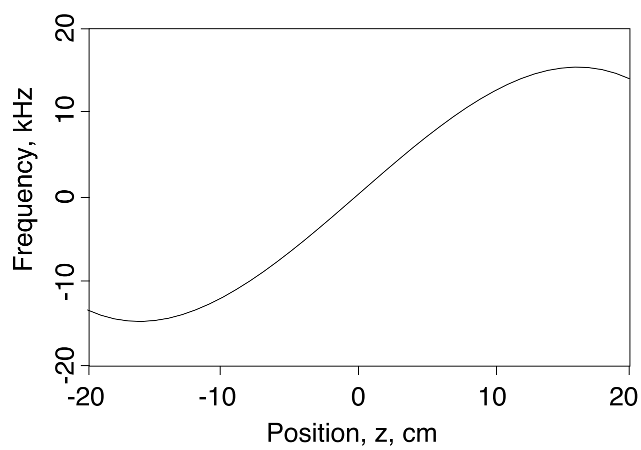
Sketch the one dimensional image we would get if we encode using the real gradient (solid line) but use the linear approximation (dashed line) when we resconstruct the data (i.e. assign spatial positions to different frequencies.) Things to look for are spatial distortion, and intensity variations.

(b) The non-linearity in the gradient can be measured, and then used to more accurately reconstruct the data. Assume that the gradient field produces a frequency

$$\omega(z) = \gamma(G_{ideal} z + H z^3) \tag{1}$$

where $G_{ideal} = 0.235$ G/cm, and $H = -1.9 \times 10^{-4}$ G/(cm^3), and z ranges from ± 20 cm. If we have a data acquisition window of 10 ms, we can resolve frequencies of 100 Hz. What spatial resolution does this provide at $z = 0,\ 10,$ and 20 cm?

(c) Would this gradient profile work for spatial encoding for MRI? Why or why not? Assume that the object extends from -20 cm to 20 cm.



5. **Matlab assignment: 2D Fourier transforms:**

Use the function `publish` to make your report.

In this assignment we will play with 2D Fourier transforms. As mentioned in class, in MRI the (0,0) coordinates both in k-space and of the image are located in the center. The functions `fft2` and `ifft2` in Matlab place the origin of the axis at the corner of the image. We will therefore need to write new functions that compute the centered DFT.

(a) Program the functions `F=fft2c(f)` and `f=ifft2c(F)`. Recall that

```
>> F = ifftshift(fft2(fftshift(f)));
>> f = fftshift(ifft2(ifftshift(F)));
```

(b) **Rectangles in k-space**. Download the functions `mysinc.m` and `myjinc.m`. The functions implement the `sinc` and `jinc` function as described in class. They accepts a vector (or matrix) of inputs and return the results in the same format as the input.

The Fourier transform of a rectangle with length 18[cm] and amplitude 1 is

$$F(k) = 18^2 \mathrm{sinc}(18k_x)\mathrm{sinc}(18k_y)$$

We would like to sample the rectangle in k-space while achieving a resolution of 1mm and a 25.6cm Field of view. Recall from class that the resolution $\Delta x = \frac{1}{W_{kx}}$ where W_{kx} is the extent in k-space. Also recall that $\mathrm{FOV}_x = \frac{1}{\Delta k_x}$ where Δk_x is the spacing between samples in k-space. For 1mm, we get that $W_{kx} = 10$ and $\Delta k_x = 10/256$.
To sample the function we will create a grid using the function `meshgrid`. First, let's create the intervals:

```
>> Wkx = 10;
>> Wky = 10;
>> Nx = 256;
>> Ny = 256;
>> kx_itvl = [ -Nx/2:Nx/2-1]/Nx * Wkx;
>> ky_itvl = [ -Ny/2:Ny/2-1]/Ny * Wky;
```

Now, we will create the grid and evaluate the function

```
>> [kx, ky] = meshgrid(kx_itvl, ky_itvl);
>> F = 18*18*mysinc(18*kx).*mysinc(18*ky);
```

Since the sample lie on a grid, we can reconstruct the data using the inverse fft. In the inverse fft, the data is scaled by $1/(NxNy) = 1/(256^2)$. On the other hand, each point in k-space represents an area of $\Delta k_x \Delta k_y = 10^2/256^2$. So, for the ifft to approximate the inverse Fourier transform we need to multiply the result by a 100. This is however not very important in MRI. The actual scaling of an image in MRI depends on many things and can vary significantly between scans. The contrast in the image is the important information. We will do it here just so the rectangle will have an amplitude of 1. But again, this is not so important.

```
>> f = 100*ifft2c(F);
>> figure(1), imshow(abs(f),[]), title('|f|')
>> figure(2), imshow(angle(f),[]), title('The phase of f');
```

Validate that the size of the rectangle is 180 pixels.

In a similar way we can sample a circle in k-space. Here's an example of a circle with a diameter of 20cm:

```
>> r = sqrt(kx.^2+ky.^2);
>> F_circ = 20^2*myjinc(20*r);
>> f_circ = 100*ifft2c(F_circ);
>> figure(1), imshow(abs(f_circ),[]), title('|f|')
>> figure(2), imshow(angle(f_circ),[]), title('The phase of f');
```

To demonstrate the modulation property of the Fourier transform we will shift the k-space of the rectangle by $2\Delta k_x$. We should expect 2 cycles of phase across the FOV in the x-axis.

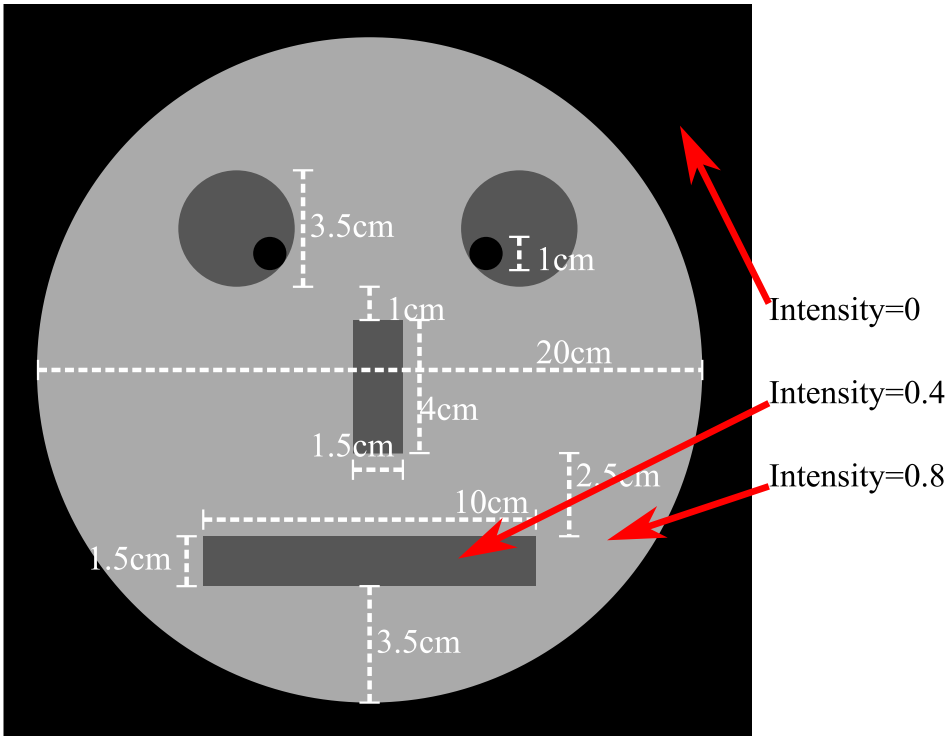
```
>> F_srect =  22*22*mysinc(22*(kx-10/256*2)).*mysinc(22*ky);
>> f_srect = 100*ifft2c(F_srect);
>> figure(1), imshow(abs(f_srect),[]), title('|f|')
>> figure(2), imshow(angle(f_srect),[]), title('The phase of f');
```

To shift a rectangle in the image domain, we need to modulate k-space accordingly. Here's an example of shifting a rectangle of length 5cm by 5cm in both axis.

```
>> F_shift =  10*10*mysinc(10*kx).*mysinc(10*ky).*exp(-i*2*pi*(5*kx+5*ky));
>> f_shift = 100*ifft2c(F_shift);
>> figure(1), imshow(abs(f_shift),[]), title('|f|')
>> figure(2), imshow(angle(f_shift),[]), title('The phase of f');
```

(c) Write the function `F = rectf(kx,ky,Lx,Ly,x0,y0)`.
The function evaluates the Fourier transform of a rectangle.
kx and ky are k-space coordinates matrices (can be generated by `meshgrid` for example).
Lx and Ly are scalars corresponding to the length of the rectangle in the x and y axis.
x0 and y0 are scalars corresponding to the coordinates of the center of the rectangle.

Write the function `F = circf(kx,ky,diam,x0,y0)`.
The function evaluates the Fourier transform of a circle.
kx and ky are k-space coordinates matrices (can be generated by `meshgrid` for example).
diam is a scaler representing the diameter of the circle.
x0 and y0 are scalars corresponding to the coordinates of the center of the circle.
Using these functions create the function `F = face(kx,ky)`.
The function evaluates the Fourier transform of the following image:



(d) Sample and display the resulting image for the following parameters:

i. FOV = 102.4 cm, resolution = 4mm

ii. FOV = 24 cm, resolution = 5mm

iii. FOV = 25.6 cm, resolution = 1mm

iv. FOV = 19.2 cm, resolution = 1mm

What happens when the imaging FOV is smaller than the object?